APPENDIX G

E.PIPHANY

E.piphany E.4 System Guide

Release 4.0

June 1999

E.piphany Confidential

E.PIPHANY E.4 SYSTEM GUIDE

RELEASE 4.0

JUNE 1999

E.piphany Confidential

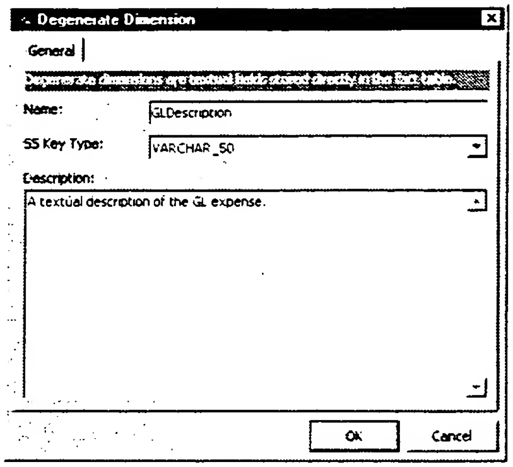
Copyright and Trademarks

This document contains propriety and confidential information of Epiphany, Inc. The contents of this document are for informational use only, and the contents are subject to change without notice. Epiphany, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Printed in the USA

Restricted Rights Legend

FIGURE 44: DEGENERATE DIMENSION DIALOG BOX



Degenerate dimensions are textual fields stored directly in the fact table.

Name: GLDescription

SS Key Type: VARCHAR_50

Description: A textual description of the GL expense.

FACT TABLES

A fact table is a physical database table that holds numeric data in its columns. It also represents the intersection of a series of dimension keys. A fact table consists of dimension role foreign keys, degenerate dimension keys, and fact columns.

An EpiCenter that utilizes Campaign Manager has additional built-in fact tables: **Ind_Group_Joiner**, **Communication**, and **SeedJoiner**. The **Ind_Group_Joiner** table specifies the association between members of the base dimensions that correspond to the `indiv` and `group` dimension roles. Each individual must belong to exactly one group. If you want to turn off the processing of individuals (for a group-only EpiCenter), you can remove the `indiv` dimension role from the **Ind_Group_Joiner** table.

The **Communication** table is the special backfeed fact table for campaigns that shows who was sent a particular mailer during a campaign. A typical row of the Communication fact table contains information about a specific treatment (promotion) applied to a single individual. This fact table captures information related to the entire campaign's communication history (current and previous marketing programs) and makes it available to the datamart for query.

The **SeedJoiner** fact table is where the seed data resides. This fact table must be present in order for seeding to occur.

Transaction Type and Date Dimensions

When defining fact tables, remember that the transtype (transaction type) and date dimensions occur in every fact table.

The transtype is a "slice" of the fact table that functions as a separate fact table. You can configure one fact table with multiple transtypes, or configure multiple fact tables. The advantage of having multiple transtypes is that measures that require two transaction types, such as `BOOK` and `BOOK_RETURN` need to issue only one query. The advantages of multiple fact tables are that queries that need only one transtype have fewer rows to navigate, and that aggregate tables are smaller.

Because the date dimension can serve as the base dimension of any dimension role, a fact table can receive more than one foreign key to the date dimension (all fact tables receive `date_key` as a foreign key). You can also define additional date dimension roles and include them in the fact table. The column name in the fact table is `dimrolename_key` (note that the suffix is not `sskey`).

Note: *Because of the special usage of* `date_key`*, it cannot be used as an attribute; use* `day_name` *as an alternative.*

As with all dates in the system, only the day-level granularity can be used for these dimension roles. The time should be 12:00 midnight during extraction, which can be accomplished by using built-in macros, such as `$$TO_EPIDATE`.

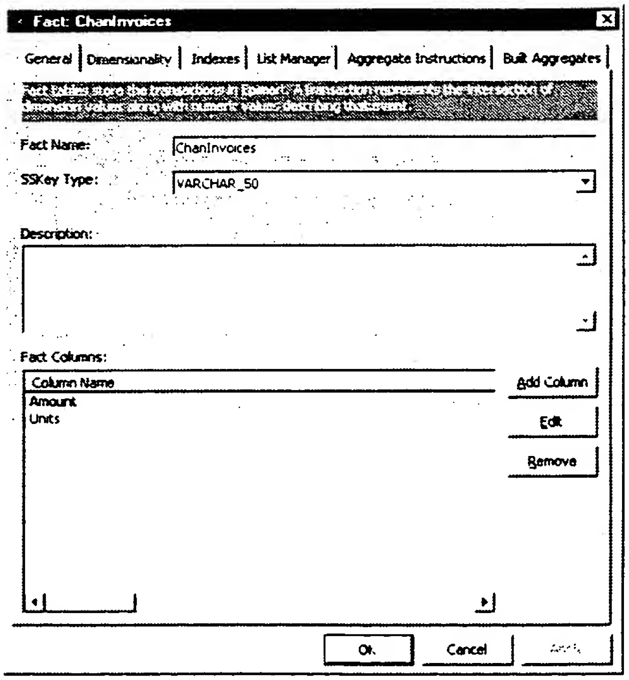
Fact semantics treat these columns in the usual way. Changes to the dimension-role value result in the booking or debooking of those fact tables that allow the changing of facts. All columns in the date dimension are available for querying. The AppServer sorts these columns in the same manner as the built-in date dimension.

DEFINING A FACT TABLE

To define a fact table:

1. Right-click the **Facts** folder icon, and select **New Fact**. The Fact Table dialog box (Figure 45) is displayed. This dialog box has tabs labeled General, Dimensionality, Indexes, List Manager, Aggregate Instructions, and Built Aggregates.
2. In the upper pane of the General tab, enter the fact table's name. The maximum length is 20 characters.
3. Select the source system key (`sskey`) data type from the drop-down list.
4. Enter a textual description of this fact table for your reference.

FIGURE 45: FACT TABLE DIALOG BOX: GENERAL TAB



Fact: ChanInvoices

Fact tables store the transactions in Epimart. A transaction represents the intersection of dimension values along with numeric values describing the event.

Fact Name: ChanInvoices

SSKey Type: VARCHAR_50

Description:

Fact Columns:

Column Name
Amount
Units

DEFINING FACT COLUMNS

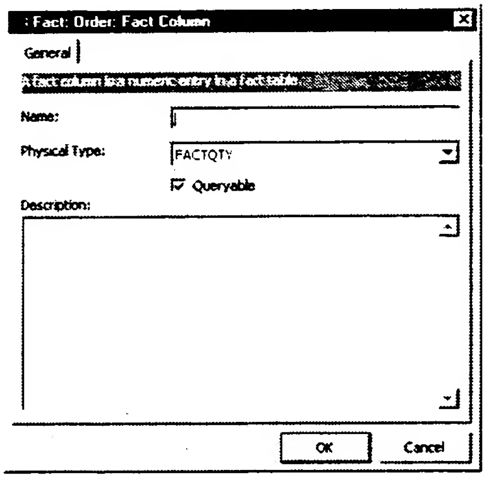
Fact columns contain the actual customer numeric data, such as `net_price` or `number_units`.

You can use the **Fact Columns** pane of the General tab of the Fact Table dialog box to define a fact column.

Follow these steps to define a fact column:

1. Click **Add Column**. The Fact Column dialog box (Figure 46) is displayed.

FIGURE 46: FACT COLUMN DIALOG BOX



: Fact: Order: Fact Column

General

A fact column is a numeric entry in a fact table.

Name:

Physical Type: FACTQTY

☑ Queryable

Description:

2. Enter the column name and select the physical type from the drop-down list. The physical type you select is replaced by its associated database type. The translation of physical type to database type depends on your RDBMS platform. See Appendix D, "Physical Type Values" for descriptions of these physical types.

Note: For this release, the FACTMONEY, FACTQTY, and EPIINT physical types are supported for fact columns.

3. The **Queryable** option is checked by default. If this option is not checked, the column cannot be moved into List Manager-related tables.

   List Manager creates composite indexes on fact tables to improve performance.

   These indexes can have no more than 16 columns in SQL Server 7.0. Exceeding this number results in degraded performance. Note the following SQL Server 7.0 restriction:

```
16 >= 2 * (ind_d_roles) + grp_d_roles + qry_dims + qry_fcts
```

Replace *ind_d_roles* with the number of individual dimension roles. Replace *grp_d_roles* with the number of group dimension roles. Replace *qry_dims* with the number of dimensions with at least one queryable attribute. Replace *qry_fcts* with the number of queryable fact columns.

4. Enter a description, and click **OK** to return to the Fact Table dialog box. The new fact column appears in the list box.
5. Repeat the above steps to define another column.

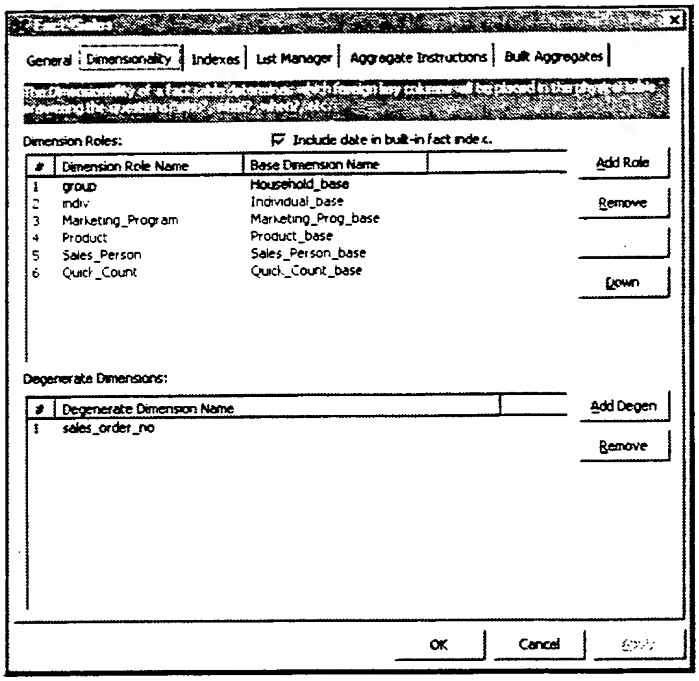
To remove a column from the fact table, select it and click **Remove**. To modify the column description, select it and click **Edit**.

FOREIGN KEYS IN FACT TABLES

Foreign keys in fact tables reference dimension tables. In a star schema, the fact table is in the center of the star, and the points that radiate outward are the dimensions (attributes stored in dimension tables). Fact tables store keys that reference these "foreign" dimension tables. There are two kinds of foreign keys: dimension roles and degenerate dimensions.

You can use the Dimensionality tab of the Fact Table dialog box (Figure 47) to assign up to 20 keys to a fact table.

FIGURE 47: FACT TABLE DIALOG BOX: DIMENSIONALITY TAB



Dimension Roles:

☑ Include date in built-in fact index.

#	Dimension Role Name	Base Dimension Name
1	group	Household_base
2	indiv	Individual_base
3	Marketing_Program	Marketing_Prog_base
4	Product	Product_base
5	Sales_Person	Sales_Person_base
6	Quick_Count	Quick_Count_base

Degenerate Dimensions:

#	Degenerate Dimension Name
1	sales_order_no

To assign a dimension role foreign key:

1. In the **Dimension Roles** pane, click **Add Role** and select one or more dimension roles from the Choose dialog box.

   Clicking the **New** button in the Choose dialog box opens the Dimension Role dialog box for creating a new dimension role.

2. Add all appropriate dimension roles.

3. Use the **Up** and **Down** buttons to order the dimension roles for the default index. Only the first dimension role (the leading term) is of significance.

   By default, each fact table has a single clustered index with the leading term of `date_key` which allows fast filtering based on time. (The leading term of an index is the most important term; for example, a phone book is indexed by last name; you cannot locate someone by first name only.)

The built-in fact tables **Communication** and **Ind_Group_Joiner** have the dimension roles `indiv` and `group` that you need to associate with the base dimension tables that contain individual and group data. Initially, these roles are set to *campaign* and *cell*, respectively.

The built-in **SeedJoiner** fact table has the dimension roles `indiv` and `group` that are already associated with the base dimension tables that contain individual (**IndivSeed**) and group (**GroupSeed**) data.

The absence of `indiv` in the **Ind_Group_Joiner** or **SeedJoiner** fact table indicates that the EpiCenter is only for a group. Thus for a group-only EpiCenter, remove the `indiv` dimension role for the **Ind_Group_Joiner** and **SeedJoiner** fact tables. The `indiv` dimension role itself, however, cannot be removed. It must always point to some base dimension table, even when it is not used.

To assign a degenerate dimension foreign key:

1. In the **Degenerate Dimensions** pane, click **Add Degen** and select one or more degenerate dimensions from the Choose dialog box.

   Clicking the **New** button in the Choose dialog box opens the Degenerate Dimension dialog box for creating a new degenerate dimension.

2. Add all appropriate degenerate dimensions.

By default, fact tables have the leading term of the index as `date_key` because a wide class of queries filters on the time dimension. This indexing allows for fast filtering based on time. You have the option of removing the `date_key` from the index. To remove it, uncheck **Include date in built-in fact index** in the Dimensionality tab of the Fact Table dialog box (Figure 47, on page 196).

If end users typically apply selective filters to other dimension roles such as customer or product, then configuring custom indexes with these leading terms improves system performance. (See “Custom Fact Indexing,” on page 92 for a more in-depth discussion of this topic.)

Custom Fact Indexes

In general, aggregates satisfy high-level queries while indexes satisfy highly selective queries. Query drill-downs on a Web page transition from broad to selective. In terms of aggregates and indexing, the sequence of a drill-down transition from top to bottom might be—

1. A small aggregate (for example, Business Unit equals Copiers).
2. A larger aggregate (month equals January 1999).
3. An index on a large aggregate (Customer Region equals West).
4. An index on a base dimension table (drill down by Customer Name equals John Doe).

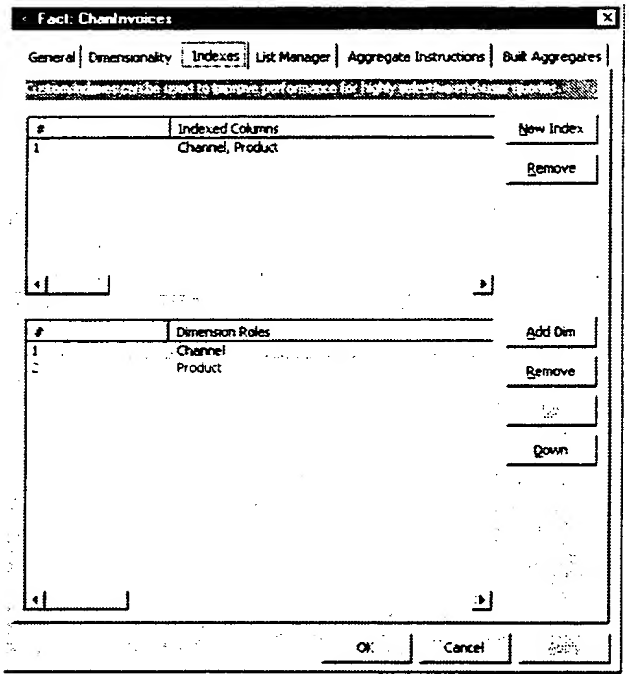
Note: *Aggregate tables are indexed in the same way as their base tables on those columns that still exist in the aggregate. Aggbuilder does not build the same index twice simply because of missing columns.*

A custom fact index is the metadata definition of an index to build on a fact table in EpiMart. It is an ordered list of dimension roles. You can use EpiCenter Manager to configure custom indexes to improve query performance significantly for selective queries.

You can use the Indexes tab of the Fact Table dialog box (Figure 48, on page 199) to create a new custom index:

1. In the **Indexed Columns** pane, click **New Index**.
2. Select one or more dimension roles from the Choose dialog box to add to this index. The first dimension role is the leading term.

FIGURE 48: FACT TABLE DIALOG BOX: INDEXES TAB



Custom indexes can be used to improve performance for highly selective end-user queries.

#	Indexed Columns
1	Channel, Product

#	Dimension Roles
1	Channel
2	Product

To modify dimension roles for an index: In the **Dimension Roles** pane, select the index and click **Add Dim** to add additional roles.

To delete the index, select it and click **Remove**.

To remove dimension roles from an index, select the role in the lower pane, and click **Remove**.

THE LIST MANAGER TAB: CLUSTERS AND COUNTS

Defining clusters for fact tables can improve performance for List Manager queries. A *fact cluster* is an index that you set up to include a significant attribute (a leading dimension role) that end users typically query for a fact table. The E.piphany system generates additional attributes for the index. For example, assume an Order fact table in your star schema has dimensions for Customers, Product Lines, Sales Regions, and Sales Force. Users typically ask about buying patterns by Customers (the demographic dimension that you have associated with the `group` dimension role) and product lines. If you defined a cluster on the dimension role that points to Product Lines, then the query machinery can bypass unrelated dimensionality, such as Sales Force, for the Order fact table.

Note: *When the* `Ind_Group_Joiner` *table serves as the fact table for transaction filters, you can define counts and clusters on it.*

A *cluster* uses the SQL Server Covered Query or the Oracle Index FFS. These queries check if the attributes for a referenced fact table are part of an index, or cluster. If so, the search engine searches that cluster instead of all of the dimension tables that the fact table points to.

Designing meaningful clusters improves performance because all of the data for the query is contained within the index itself. Therefore, only the index pages, not the actual tables, must be referenced to retrieve the data.

Counts keep statistics about how often a row appears in the fact table. Counts are used by the List Manager query engine to select the best cluster for the fact table.

Note: *Counts are used on Oracle databases to make queries run faster. Counts defined for SQL Server are ignored.*

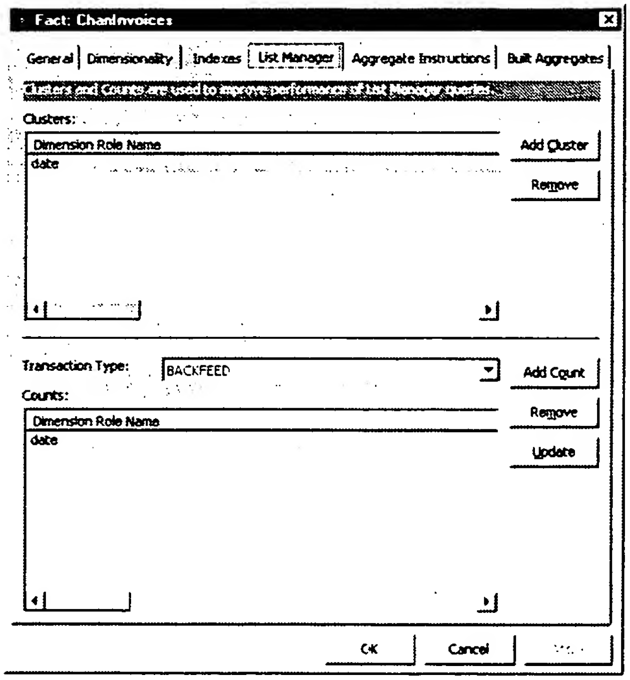
Guidelines for counts and clusters:

- You need to specify at least one cluster on a fact table for it to be available to the List Manager tables (using the List Manager tab of the Fact Table dialog box).

- To be able to cluster a fact table, you need to specify a dimension role that corresponds to a dimension other than a demographic dimension.
- *Oracle only*: Counts are associated with transaction types. Create a separate count for each transaction type associated with the fact table for this cluster. *Oracle does not use a cluster unless there is an associated count for the transaction type being queried.*
- Because cluster usage has some system overhead, whereas counts do not, be judicious in your use of clusters.

The List Manager tab of the Fact Table dialog box allows you to define clusters and counts.

Figure 49: Fact Table Dialog Box: List Manager Tab



Fact: ChanInvoices

Clusters and Counts are used to improve performance of List Manager queries.

Clusters:

Dimension Role Name
date

Transaction Type: BACKFEED

Counts:

Dimension Role Name
date

Follow these steps to define a Cluster:

1. In the **Clusters** pane, click **Add Cluster**.
2. Choose a non-demographic dimension role for this cluster from the Choose dialog box. This dimension role should be the leading term.

(*Oracle only*) To define a count for this cluster:

1. In the **Counts** pane of the List Manager tab, click **Add Count**.
2. Select one of the associated transaction types for this fact table from the drop-down list. Click **OK**.
3. Choose the dimension role for the count from the Choose dialog box.
4. Repeat these steps to create a separate count for each transaction type associated with this fact table and cluster.

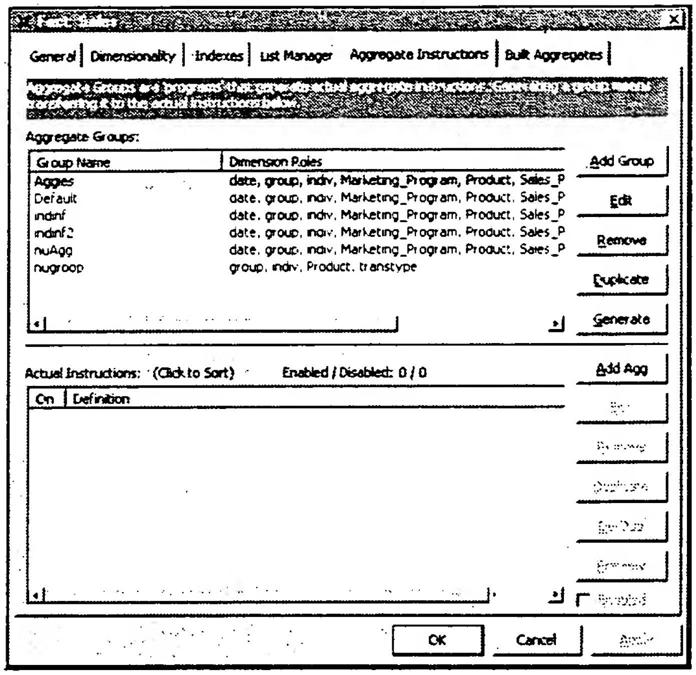
DEFINING FACT AGGREGATES

The Aggregate Instructions tab of the Fact Table dialog box (Figure 50) allows you to define fact table aggregation. The dimensions you choose to aggregate depend on which of the fact's dimensions end users might typically query as a group. EpiCenter Manager uses your specifications to generate the actual instructions in metadata that control how Aggbuilder builds aggregates for the fact table.

Having the system aggregate fact values for these dimensions in advance speeds up front-end query time. In general, when designing fact aggregates, attempt to strike a balance between broad fact aggregates (some compression/high query coverage) and narrow fact aggregates (high compression/low coverage).

To configure instructions for the fact table, open the Aggregate Instructions tab (Figure 50, on page 203).

FIGURE 50: FACT TABLE DIALOG BOX: AGGREGATE INSTRUCTIONS TAB



Aggregate Groups are programs that generate actual aggregate instructions. Generating a group means transferring it to the actual instructions below.

Aggregate Groups:

Group Name	Dimension Roles
Agges	date, group, indiv, Marketing_Program, Product, Sales_P
Default	date, group, indiv, Marketing_Program, Product, Sales_P
indinf	date, group, indiv, Marketing_Program, Product, Sales_P
indinf2	date, group, indiv, Marketing_Program, Product, Sales_P
nuAgg	date, group, indiv, Marketing_Program, Product, Sales_P
nugroop	group, indiv, Product, transtype

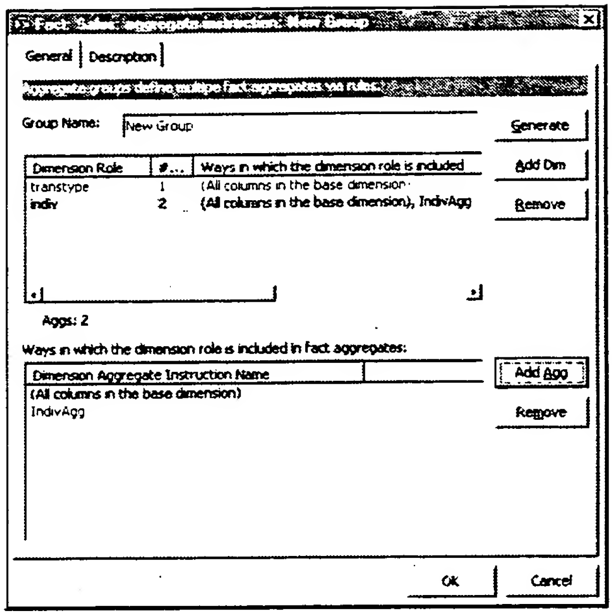
Actual Instructions: (Click to Sort) Enabled / Disabled: 0 / 0

On	Definition

Follow these steps to create aggregate groups:

1. Click **Add Group**.
2. In the General tab of the New Group dialog box (Figure 51, on page 204), enter the name of the group.
3. You can open the Description tab at any time to enter a description of this group.

FIGURE 51: NEW GROUP DIALOG BOX



Aggregate groups define multiple fact aggregates via rules.

Group Name: New Group

Dimension Role	#...	Ways in which the dimension role is included
transtype	1	(All columns in the base dimension·
indiv	2	(All columns in the base dimension), IndivAgg

Aggs: 2

Ways in which the dimension role is included in Fact aggregates:

Dimension Aggregate Instruction Name	
(All columns in the base dimension)	
IndivAgg	

4. The fact aggregates that can be generated from an aggregate group are determined by a combinatorial expansion based on the dimension role possibilities that you set up. Click **Add Dim** and select a dimension role for this group from the Choose dialog box.
5. All of a selected dimension role's base table columns display in the lower pane of the dialog box, in addition to the inclusive **All columns in the base dimension**. You can specify which columns are included in the aggregate by deleting those you do not want included. Select them in the lower pane and click **Remove**.
6. Add additional dimension roles for the group and specify the columns to be included.

   The total number of aggregates for the group is shown on the tab.
7. Click OK after you have defined the new group. The group is added to the listing in the upper pane of the Aggregate Instructions tab.
8. Create additional groups as described above. To add a similar group, select it and click the **Duplicate** button. Click **Edit** to modify the group.

After you have defined the groups, you can generate the actual instructions in metadata by clicking **Generate** in the Aggregate Instructions tab. Generating transfers the aggregate instructions to the **Actual Instructions** shown in the lower pane (Figure 50).

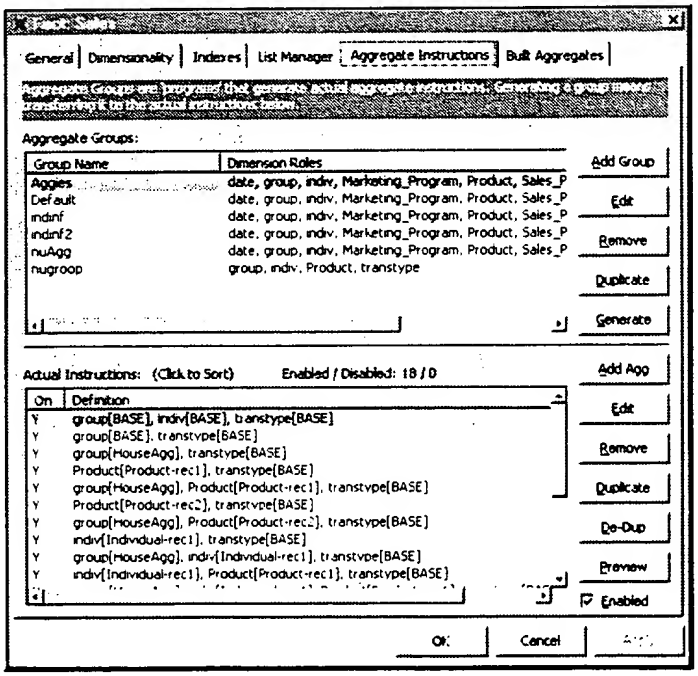
After you have added all of the instructions, click **De-Dup** to remove any duplicates that may have been created.

To sort instructions by dimension roles, click an item in the listing.

To disable aggregation for this fact table, uncheck **Enabled**.

Clicking the **Preview** button in the Aggregate Instructions tab opens the Aggregate Optimizer dialog box (Figure 36, on page 179).

FIGURE 52: AGGREGATE INSTRUCTIONS GENERATED



Aggregate Groups:

Group Name	Dimension Roles
Aggies	date, group, indiv, Marketing_Program, Product, Sales_P
Default	date, group, indiv, Marketing_Program, Product, Sales_P
indinf	date, group, indiv, Marketing_Program, Product, Sales_P
indinf2	date, group, indiv, Marketing_Program, Product, Sales_P
nuAgg	date, group, indiv, Marketing_Program, Product, Sales_P
nugroop	group, indiv, Product, transtype

Actual Instructions: (Click to Sort) Enabled / Disabled: 18 / 0

On	Definition
Y	group[BASE], indiv[BASE], transtype[BASE]
Y	group[BASE], transtype[BASE]
Y	group[HouseAgg], transtype[BASE]
Y	Product[Product-rec1], transtype[BASE]
Y	group[HouseAgg], Product[Product-rec1], transtype[BASE]
Y	Product[Product-rec2], transtype[BASE]
Y	group[HouseAgg], Product[Product-rec2], transtype[BASE]
Y	indiv[Individual-rec1], transtype[BASE]
Y	group[HouseAgg], indiv[Individual-rec1], transtype[BASE]
Y	indiv[Individual-rec1], Product[Product-rec1], transtype[BASE]

☑ Enabled

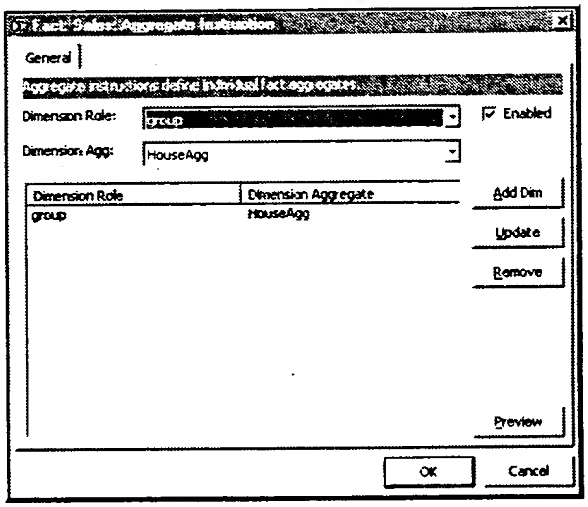
DEFINING/EDITING AGGREGATE INSTRUCTIONS

You can define an aggregate instruction by clicking **Add Agg** in either the **Aggregate Instructions** pane of the Aggregate Instructions tab, or the New Group dialog box. In the **Aggregate Instruction** dialog box (Figure 53) that appears, specify the dimension aggregate. Follow these steps:

1. Select the dimension role from the drop-down list.
2. Select the dimension aggregate.
3. Add any other dimension roles/dimension aggregates.
4. Click **OK** to add this aggregate instruction to the list of other defined instructions.

You can edit an aggregate instruction by double-clicking it, or by selecting it and clicking **Edit**, which displays the **Aggregate Instruction** dialog box (Figure 50).

FIGURE 53: AGGREGATE INSTRUCTION DIALOG BOX



Aggregate instructions define individual fact aggregates.

Dimension Role: group

Dimension Agg: HouseAgg

☑ Enabled

Dimension Role	Dimension Aggregate
group	HouseAgg

REMOVING/DISABLING AGGREGATE INSTRUCTIONS

An aggregate group often produces some instructions that are infrequently used in actual user queries. Removing or disabling these can significantly reduce extraction time and disk usage.

If an aggregate instruction is not longer needed, you should disable it, instead of removing it. Removing the instruction in this one case might not remove it completely from the system, whereas disabling turns off all occurrences of the instruction. To disable an aggregate instruction, open the instruction's Aggregate Instruction dialog box and uncheck **Enabled**.

BROWSING FACT AGGREGATES

As part of refining your EpiCenter, you may wish to browse the list of fact aggregates that Aggbuilder built according to your instructions to determine which tables are available for the query machinery. You can use the Built Aggregates tab of the Fact Table dialog box (Figure 54) for this purpose.

You can browse the A/B/X/Y sets of tables as appropriate. Select the set from the drop-down list under the **Filter** button. There are four copies of every fact table, with suffixes _A, _B, _X, and _Y. The _A and _B tables are mirrored copies of a table containing recent facts, and their operation is similar to that of the _A and _B dimension tables. The _X and _Y tables are mirrored copies of a table containing historical (or less-recent) facts. At any time, either the X or Y history table is active. If the active tables are set to A and X, then all end-user queries that refer to a fact table are run against the union of the versions of that fact table that have A and X suffixes. The B and Y tables contain data that is one extraction older than the A and X tables, and they are inactive. The current datamart is shown in the lower-right corner of the EpiCenter Manager window.

You can also see which fact aggregates were *not* built for selected dimension roles. To do so:

1. Select **Not Included** from the **Dimension Aggregate** drop-down list.
2. Select a dimension role in the upper pane, and click **Filter**.
3. View the listing in the lower pane.

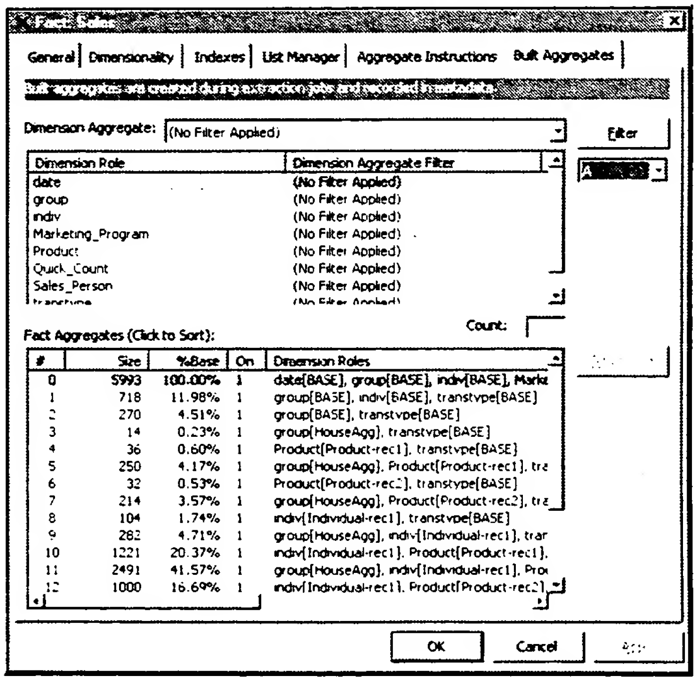
The **Fact Aggregate** pane shows the following information:

- The total number of fact aggregates built (with the current filter applied) is shown in the **Count** box.
- The fact aggregate number.

  Although this numbering is arbitrary, it may be helpful when debugging query logs since the logs use these numbers.
- The **Size** and **%Base** (number of rows in this aggregate divided by the number of rows in the base fact table expressed as a percentage).

  You can click the **Size** or **%Base** headings to sort aggregates by size. This sorting shows which aggregates are most valuable (a larger percentage means greater aggregation).
- The associated dimension roles.

FIGURE 54: FACT TABLE DIALOG BOX: BUILT AGGREGATES TAB



Built aggregates are created during extraction jobs and recorded in metadata.

Dimension Aggregate: (No Filter Applied)

Dimension Role	Dimension Aggregate Filter
date	(No Filter Applied)
group	(No Filter Applied)
indiv	(No Filter Applied)
Marketing_Program	(No Filter Applied)
Product	(No Filter Applied)
Quick_Count	(No Filter Applied)
Sales_Person	(No Filter Applied)
transtype	(No Filter Applied)

Fact Aggregates (Click to Sort):

Count:

#	Size	%Base	On	Dimension Roles
0	5993	100.00%	1	date[BASE], group[BASE], indiv[BASE], Marke
1	718	11.98%	1	group[BASE], indiv[BASE], transtype[BASE]
2	270	4.51%	1	group[BASE], transtype[BASE]
3	14	0.23%	1	group[HouseAgg], transtype[BASE]
4	36	0.60%	1	Product[Product-rec1], transtype[BASE]
5	250	4.17%	1	group[HouseAgg], Product[Product-rec1], tra
6	32	0.53%	1	Product[Product-rec2], transtype[BASE]
7	214	3.57%	1	group[HouseAgg], Product[Product-rec2], tra
8	104	1.74%	1	indiv[Individual-rec1], transtype[BASE]
9	282	4.71%	1	group[HouseAgg], indiv[Individual-rec1], trar
10	1221	20.37%	1	indiv[Individual-rec1], Product[Product-rec1],
11	2491	41.57%	1	group[HouseAgg], indiv[Individual-rec1], Pro
12	1000	16.69%	1	indiv[Individual-rec1], Product[Product-rec2],

TRANSACTION TYPE SETS

Transaction types, such as Booked and Booked Returned, are defined in the Transaction Types tab (Figure 114, on page 420) of your Configuration dialog box. You are given a group of generic transaction types, and can use this dialog box to define other transaction types.

These transaction types can be grouped in a set and applied to a measure term. Measure terms, which are discussed in "Measure Terms," on page 248, define the steps that determine how a measure that an end user selects on a Web page is calculated. A *measure term* is one component of an arithmetic expression that makes up a measure. It refers to the aggregation of a single fact column in a fact table, such as `SUM(Order.net_price)` with a transaction type, or a set of transaction types.

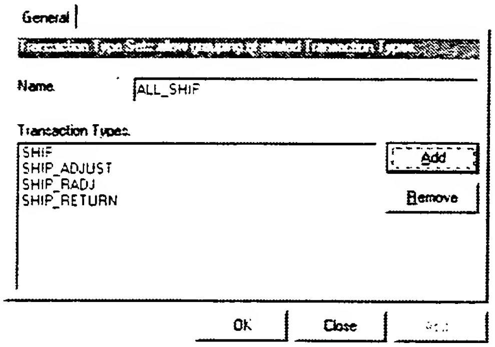
Because measure terms created via EpiCenter Manager specify a transaction type in addition to a fact column, only those rows in a fact table with the same transaction type can be summed. Applying a set of transaction types instead of individual transaction types enables the use of a single measure term calculation (usually `SUM`) to add up all rows for all transaction types in the set.

You can use the Transaction Type Sets dialog box to group transactions into sets.

To define a transaction type set:

1. Right-click the **Transaction Type Set** folder icon and select **New Transaction Type Set**.
2. Enter the name of the set in the Transaction Type Set dialog box (Figure 55, on page 210).
3. Click **Add**. Select the transaction types in the set from the Choose dialog box.

FIGURE 55: TRANSACTION TYPE SET DIALOG BOX



Transaction Type Sets allow grouping of related Transaction Types

Name ALL_SHIF

Transaction Types
SHIF
SHIP_ADJUST
SHIP_RADJ
SHIP_RETURN

This completes the instructions for defining your star schema. The next section, Extraction, describes how to configure the metadata for jobs that move the source data into these schema tables.

EXTRACTION

After you have configured your EpiCenter, you may use the **Extraction** folder to set up the extraction jobs. The **Extraction** folder consists of the sub-folders **Data Stores**, **Jobs**, **External Tables**, **Macros**, and **Queues**, each of which is discussed below.

DATA STORES

A data store is a logical location of data to be used either as a source (input data store) or sink (output data store) within an EpiCenter. Typically, the input data store is a source system database or file, and the output data store is EpiMart, which holds customer data, on an RDBMS server.

E.piphany provides two special default datamart stores: `Epimeta` and `Epimart`. `Epimart` is the typical datamart store. There are two default log stores: `LoggingDB` and `JobFile` Log. Associated with data stores are the default data store roles: `Input`, `Logging`, `Output`, and `WorkingDir`.

You can use the Data Store dialog boxes to create as many data stores as are appropriate for your site. The default data stores and any data store you create using the Data Store dialog box are available for selection in the **Object Gallery** when you define jobs.

The job dialog box is where you assign data stores and roles for an entire job. You can also assign data stores to individual job steps. For example, an extractor job has a single input and output data store. Because jobs consist of multiple extractors, a job has multiple input and output data stores. For example, one job might extract data from two source systems into one EpiMart, and another job might extract data from a source system into external tables and from external tables into staging tables. In the second job, the external tables serve as both input and output data stores to different extractors.

THE DATA STORE DIALOG BOX

Each data store is represented by a Data Store dialog box (see Figure 56). This dialog box has tabs named General and Properties.

The **Data Store Type** pane of the General tab is where you enter the name and description for the data store and select its type: SQL Server, Oracle, Generic ODBC Data Source, or File.

Note: *The **Log** and **Datamart** options in the top-right corner are read-only and simply indicate when the opened data store is one of the built-in data stores.*

The **Data Flow** pane of the General tab has the following options:

- **Allow Use as Input Data Store** allows the data store associated with an extractor job step to be used as the input of an extractor.

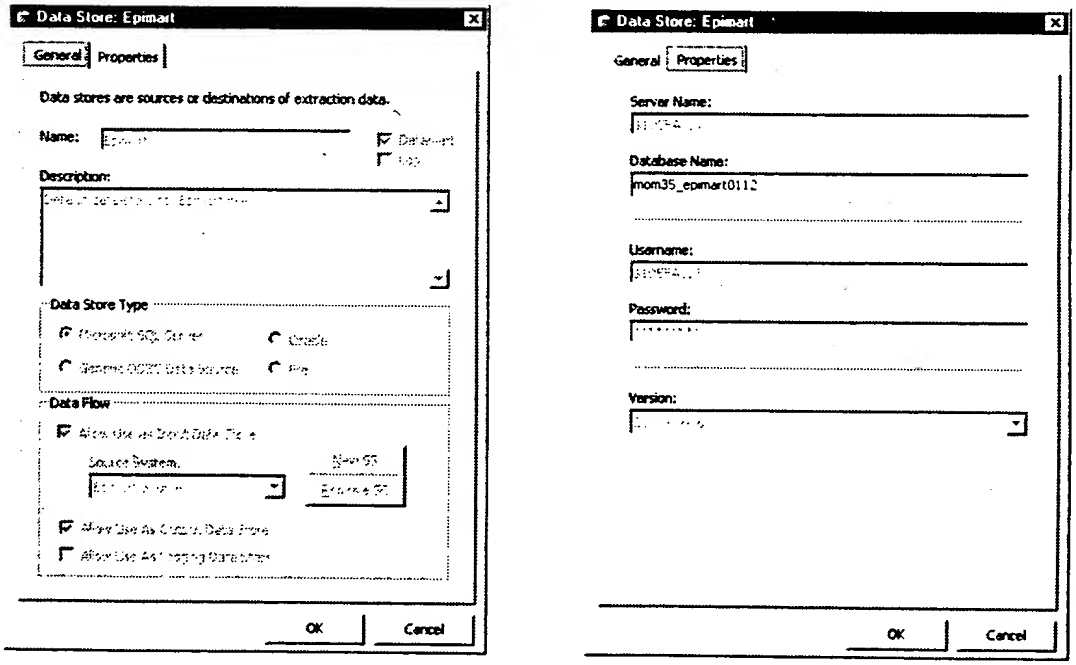
  The source systems listed in the Source System drop-down list serve to distinguish source system identifier keys that may clash between different database systems. For example, if an account has an ERP and an SFA system, each of which has a Customer table, both may contain a customer number 100. To distinguish between these two records, two different source system identifiers must be used.

Note: *Fact rows join only with dimension rows that have the same source system identifier. For this reason, unless the site actually uses two or more logical source systems, select* ***Datamart Source*** *from the* ***Source System*** *drop-down list (the default).*

- **Allow Use as Output Data Store** allows the data store to be used as the output of a job step.

  If this option is unchecked, an error message displays if the user tries to attach this data store.

FIGURE 56: DATA STORE DIALOG BOX: GENERAL AND PROPERTIES TABS



Data Store: Epimart

General | Properties

Data stores are sources or destinations of extraction data.

Name: [illegible]

☑ Datamart
□ [illegible]

Description: [illegible]

Data Store Type

● Microsoft SQL Server ○ Oracle
○ Generic ODBC Data Source ○ File

Data Flow

☑ Allow Use as Input Data Store

Source System: [illegible]

New SS
[illegible]

☑ Allow Use As Output Data Store
□ Allow Use As Staging Data Store

OK Cancel

Data Store: Epimart

General | Properties

Server Name: [illegible]

Database Name: mom35_epimart0112

Username: [illegible]

Password: ·········

Version: [illegible]

OK Cancel

- **Allow Use as Logging Data Store**

  This option ensures that logging is directed to the proper logging data store and must be selected in the Data Store dialog box.

  E.piphany has special metadata tables for the purpose of logging its activity. Normally, this logging is written to the EpiMeta database. Logging to the EpiMeta, however, may increase its size significantly, so you may prefer to log to another data store. You can use the `logging_mssql.sql` script (included with the E.piphany software) to create a new logging database. If you run this script on another database, then a new data store can be used to log extractions.

The Properties tab has different fields, depending on the data store type. You can use the Properties tab to enter these fields:

- For SQL Server, enter the server's name, database name, username, and server's password and version.
- For Oracle, enter the Net8 name, username, the server's password, and the version number (Oracle8).
- For a generic ODBC data store, enter the DSN name and ODBC driver name, and server administrator's username and password. (You must also set up a DSN for the data store system using the Data Source Administrator in the Windows NT Control Panel.)
- For a data store of type file (`JobLogFile`), enter its file directory path.

MODIFYING THE DEFAULT DATA STORES

You need to modify the default data stores, which are `Epimart`, `Epimeta`, `JobFileLog`, and `LoggingDB`. Double-click their folders to open them and fill out the dialog boxes as follows:

- **Epimart** specifies your EpiMart as a data store.

  The only value that you can configure is the name of your EpiMart. You can use the Properties tab of the EpiMart Data Store dialog box to enter this database name, for example **Corp_EpiMart**.

- **Epimeta** is the data store that references your current data, EpiMeta.
- **JobFileLog** specifies the data store for EpiChannel job logs.

  In the General tab, the settings are correct for the default usage. The Data Source Type is a file, and the Data Flow is set to **Allow Use as Logging Data Store**.

  Click the Properties tab, and change the directory for the `JobFileLog` from `CHANGE ON INSTALL` to the correct directory name. A valid directory name is required for a successful extraction. Leave the file name blank.
- **LoggingDB** specifies the data store for the EpiChannel logs.

  In the General tab, the Data Source Type is your server, and Data Flow specifies: **Allow Use as Input Data Store** and **Allow Use as Logging Data Store**.

  You can accept the default values for the log database's file name and directory path, unless you have located the log in a different database.

Note: *`$$DEFAULT` refers to the current EpiMeta.*

SETTING UP JOBS

The Job dialog box enables you to define jobs, create and order the job steps (extractors and system calls), assign data stores and roles, and schedule the job in a queue.

An EpiCenter has three default jobs–Incremental, Initial, and Merge–which are similar except for the semantics. The default jobs consist of the following:

- **Truncation**: a group with truncation steps for all of the built-in tables. You need to add truncation steps for the rest of the tables.
- **Default Extraction**: A group with empty groups for the fact and dimension table extraction.
- ***job_type*** **Semantics**: A group with empty groups for the dimension and fact semantics. There are groups for the Incremental, Initial, and Merge job types.

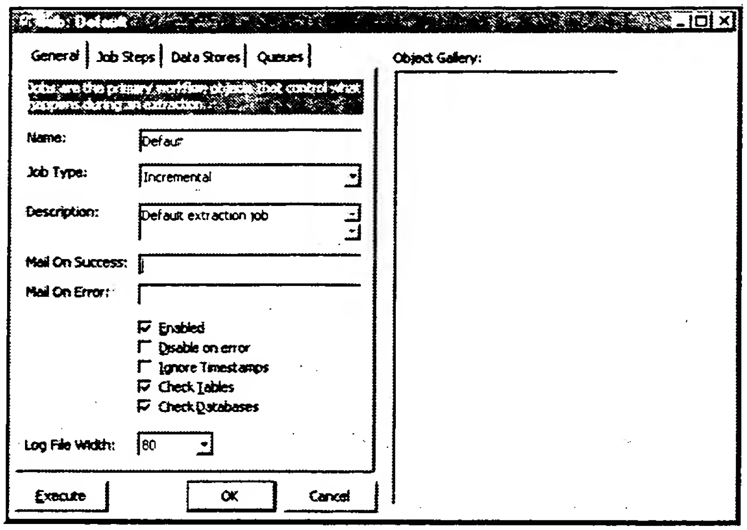
- **Campaign Extraction**: A group with SQL-extraction steps and semantics for the backfeed tables (to move data from the backfeed tables into the datamart). You do not need to modify these SQL extraction steps unless one of the campaign-related tables has been changed.
- **AggBuilder**
- **MomBuilder**
- **e.4 End of extraction**, which consists of the following:
  - **MaxDate**–an empty group to which you need to add SQL to extract the date.
  - **Max of Maxdate**–SQL that finds the highest value extracted by MaxDate and records it as the last extract date.
  - **Campaign backfeed**–a group with special semantics that update the backfeed tables.
  - **Hotswap**–toggles the A/B/X/Y tables.
- **Refresh**: The system call to refresh the AppServer.

You may open a default job, rename it, and customize it, as well as create as many other jobs as necessary. To open a default Job dialog box, double-click it.

To open a new job dialog box, right-click the **Jobs** folder and select **New Job.**

The Job dialog box has four tabs: General, Job Steps, Data Stores, and Queues.

FIGURE 57: JOB DIALOG BOX: GENERAL TAB



Jobs are the primary workflow objects that control what happens during an extraction.

Name: Defaur

Job Type: Incremental

Description: Default extraction job

Mail On Success:

Mail On Error:

☑ Enabled
□ Disable on error
□ Ignore Timestamps
☑ Check Tables
☑ Check Databases

Log File Width: 80

The General tab provides options you can set that control the following aspects of a job:

- For a job other than the default one, assign a name and description.
- Select a job type from the drop-down list. The options are described in “Job Types” on page 218.
- Assign addresses for e-mail notification of the job’s success or failure. See “Configuring E-Mail,” on page 243.
- Enable a job. EpiChannel executes enabled jobs only.

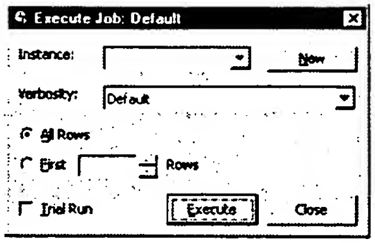
  New jobs are initially enabled. You may, however, disable a completely functional job in some circumstances to accommodate system changes. For example, if a database is in the process of being moved or repaired, jobs that populate that database could be disabled as a protection against accidental execution.
- Request that the job be disabled if it encounters an error during execution.

- Request that all timestamps be ignored during job execution (**Ignore Timestamps**).

  For example, by ignoring timestamps, you could retrieve all rows, without any date filters (based on EpiChannel's special filtering used for incremental extractions).
- Request that before executing the job, EpiChannel check that all databases referenced by any job databases and all EpiMart tables are available. Unless you are sure that this is true, select this option.
- Select the width of the log, either 80 or 132 columns. Select **80** for VGA monitors and **132** for SVGA monitors.
- Execute a single job.

  Click the **Execute** button in the Job dialog box to execute this job. You can use the Execute Job dialog box (Figure 58) to do the following:
  - Run the job as a trial run.
  - Normally, the proper instance name is already selected. You can enter a new one by clicking the **New** button.
  - Select a debug level that corresponds to the EpiChannel (**extract.exe**) verbosity levels (although you cannot specify row numbers). This debug level takes effect before the first SQL statement.
  - Indicate the total number of rows to be transferred in a pull/push operation: all rows or the first N rows. This value is reset with each extraction statement.

FIGURE 58: EXECUTE JOB DIALOG BOX



Execute Job: Default

Instance:

Verbosity: Default

● All Rows

○ First Rows

□ Trial Run

Using the **Execute** button is equivalent to invoking the **extract.exe** command with this job as the job option. The **Execute** button, however, also supplies the database name, password, and so forth, if necessary (no instance is supplied) so that **extract.exe** does not depend upon the Registry entries.

JOB TYPES

Job types control how data is merged between the historical and current partitions of a fact table. Select one of these job types for each job:

- Initial

  The Initial job type clears both historical and current partitions, and then allows you to load all newly extracted data into the historical partition. Select this type only when you are populating a newly initialized datamart, or replacing all of the data in an existing datamart. All data is moved into the historical partition, and the current partition is cleared. Lists are invalidated.

- Incremental

  This job type allows you to add data to the current partition only. You cannot make changes to the historical partitions. You can add new or updated data to the current partition, although you cannot change any data that is already in the partition. Incremental jobs improve performance by reducing the copying of historical data. In addition, existing aggregates are updated only from the contents of current partitions. (See "Aggregate Building" on page 82 for more information about aggregates.)

- Merge

  The Merge job type moves data from a current partition to a historical partition. New or updated dimension rows can be added, but no dimension rows can be changed. If historical fact data is changed, then aggregates are completely rebuilt. Otherwise, aggregates are updated without a complete rebuild.

- Rebuild

  The Rebuild job type allows you to make changes to fact and dimension tables in both current and historical partitions. All data is moved into the historical partition and the current partition is cleared. Aggregates are completely rebuilt.

- Non-Partitioned

  Does not make use of the historical partitions. All data is kept in the current partition. Changes can be made to both fact and dimension tables. Do not assign this job type for a datamart that utilizes historical partitions, as all historical data is deleted. Aggregates are completely rebuilt.

JOB STEPS

The Job Steps tab is the main dialog box for job definition. In the right window of the dialog box you can define global objects for job steps and organize them into groups. In the left window, you can define the steps for a job. By selecting objects and dragging them onto job steps, you define the actual steps of the job.

The way you use this dialog box is described below:

1. Define a job step by clicking **New Step** in the Job Steps window of the dialog box. Enter the label name for a step and any description. (Job steps that you create locally appear in boldface, global extraction groups and nodes do not.) Click **OK**.

   You can define a step but not have it execute by selecting it and clicking the **Disable** button, which toggles between **Disable** and **Enable**. A disabled step is identified by an **X** over its icon. To enable a disabled step, select it and click **Enable**.

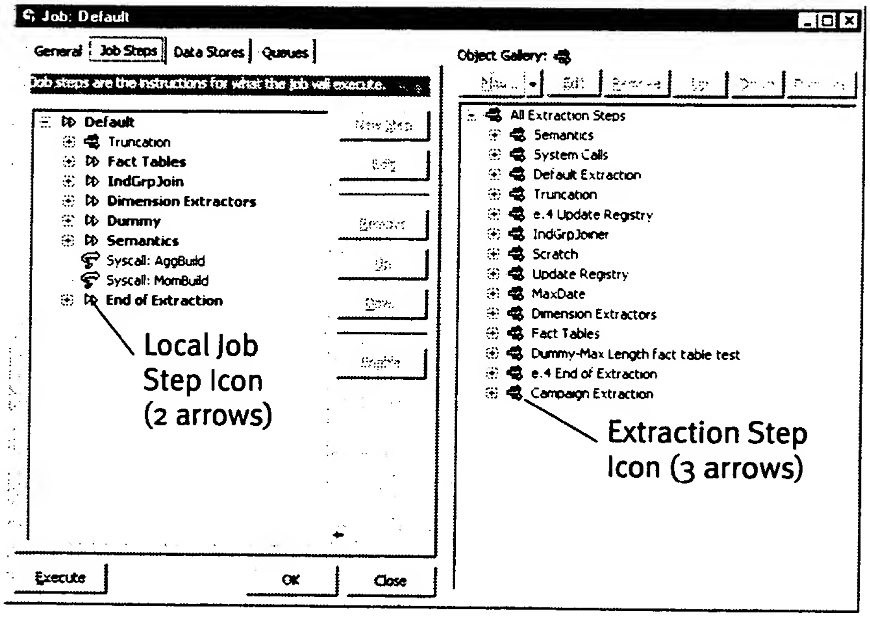
2. Order steps by selecting them and clicking the **Up** or **Down** buttons.
3. Clicking **Remove** deletes a selected step.
4. Click **Edit** to open a selected step in order to change its name and description.

5. After defining objects in the **Object Gallery**, you can select and drag them onto the job step in the left window for inclusion.

   The **Object Gallery** already contains the objects for the default job.

Note: *A local job step icon consists of two arrows; an extraction step object icon consists of three arrows (see Figure 59).*

FIGURE 59: JOBS DIALOG BOX: JOB STEPS TAB



Local Job Step Icon (2 arrows)

Extraction Step Icon (3 arrows)

To define a new global object:

1. Use the **New** drop-down list in the right window of the dialog box to define groups into which you can place related job objects: SQL, semantic instance, system call, and truncation. All of these objects are at the same level.
2. After you have created an object, you may select it in the **Object Gallery** and click **Edit** to modify its dialog boxes, or click **Remove** to delete it. These changes affect all occurrences of the object in the EpiCenter.
3. You can also select an object and click **Up** or **Down** to reorder it in the list.
4. Clicking **Duplicate** opens a dialog box for you to enter a new name for a copy of the object's definition. Double-click the new object name to open its dialog box for editing.

Note: You can disable individual parts of a global extraction group. These items are disabled for this job only.

DEFINING A NEW GROUP

To define a new group:

1. In the **Object Gallery**, either select a group (groups may contain groups), or select **All Extraction Groups** to place the group at the top level.
2. Hold down the arrow to the right of the New button and choose **Group** from the drop-down list.
3. In the dialog box, enter the group name and any description.
4. Indicate the action to be taken upon error (abort, ignore, or print).
5. After clicking **OK**, the new group is displayed as an object in the Gallery.

A group can contain other groups.

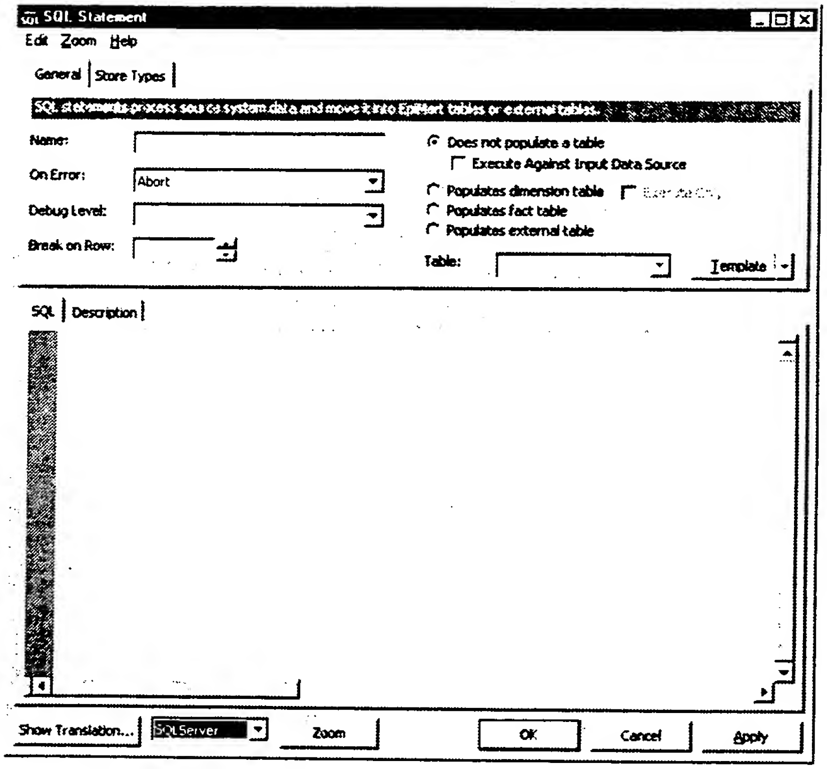
DEFINING SQL

If a job step requires SQL, EpiCenter Manager provides a template for the sample SQL. SQL statements process source system data and move it into EpiMart or external tables.

To define an SQL object:

1. Select the group for the new SQL in the **Object Gallery**.
2. Choose **SQL** from the **New** drop-down list. The SQL Statement dialog box (Figure 60) is displayed. It has two tabs: General and Store Types.

FIGURE 60: SQL STATEMENT DIALOG BOX: GENERAL TAB



SQL statements process source system data and move it into EpiMart tables or external tables.

Name:

On Error: Abort

Debug Level:

Break on Row:

● Does not populate a table

□ Execute Against Input Data Source

○ Populates dimension table □

○ Populates fact table

○ Populates external table

Table:

In both tabs, the same SQL and Description tabs are in the lower pane. (You can use the description tab to add a description for your reference.)

3. In the upper pane of the General tab, enter a name for the SQL step, such as a step named **Customer Raw** for extracting raw customer data. When the step is added to the **Object Gallery, SQL:** is placed before this name to identify it as SQL.
4. Select the action to be taken if there is an error with the step. The default is to abort the step. (You can also ignore the error, or have it print to a log and continue.)
5. Select the debug level. These levels correspond to the verbosity levels on the **extract.exe** command line. See "EpiChannel Debugging Levels," on page 116.

6. If you select a row number in Break on Row, execution stops at that row number for debugging purposes. If set to 0, the entire statement runs.

Note: *You may set breakpoints on SQL statements that change the debug level before the SQL is issued. See "Setting Breakpoints," on page 117.*

7. Select whether the step:

   - populates one of the following types of tables: a dimension table, a fact table, or an external table, or
   - does *not* populate a table (and if this is true, select whether it executes against input data store, or retains its default behavior of executing against the destination data store).

   Most SQL statements are used to populate staging tables (or sometimes, external tables). An SQL statement may, however, be used to achieve a side-effect, in which case you can set it to "not populate a table." In this case, the statement is executed, and any returned results discarded.

8. If the step references a table, select the table from the drop-down list.

   Select this when SQL is to be expanded to reflect the structure of one of these tables, but the returned rows from the SQL statement are not meaningful. EpiChannel does not insert rows in the statement.

Extraction SQL is executed against the input data store and must be in the dialect of that database engine. The results are stored in the destination data store. EpiChannel automatically resolves SQL dialect problems if you use the E.piphany database-vendor independent macros consistently in your SQL. (See Appendix A, "E.piphany Macros" for more information.)

DISPLAY SAMPLE SQL

If the step populates a table, clicking the **Template** button displays sample SQL for the step in the lower pane of the SQL Statement dialog box. The SQL needs to be modified to contain the FROM and WHERE clauses appropriate for the tables in the source system, but the template lists the columns that must be returned. It does not matter in what order the columns are returned as long as they have the proper column names (and "extra" unused columns may be returned).

You can also view sample template code for a single column in a table. Holding down the small arrow to the right of the **Template** button opens a menu of the columns in the target table. Select the column whose template code you want to display. Select **All** to display code for all of the table's columns.

You can have the system show the macro's translation for your server. Click the **Show Translation** button and select your server type from the drop-down list.

The SQL Results window displays the SQL in color-coded text. The main menu commands enable you to use standard edit commands to search and replace lines, and you can set bookmarks on lines. The **Tools** menu has **Undo** and **Print** commands, and commands for clearing text with and without clearing the buffer.

You can resize the window, and click the **Zoom** button to have the text area fill up the entire SQL or Description window.

RESTRICT DATA STORES FOR THE EXTRACTION

You can restrict the input/output data stores for this extraction to Oracle and SQL Server. The statement is disabled for any unchecked type.

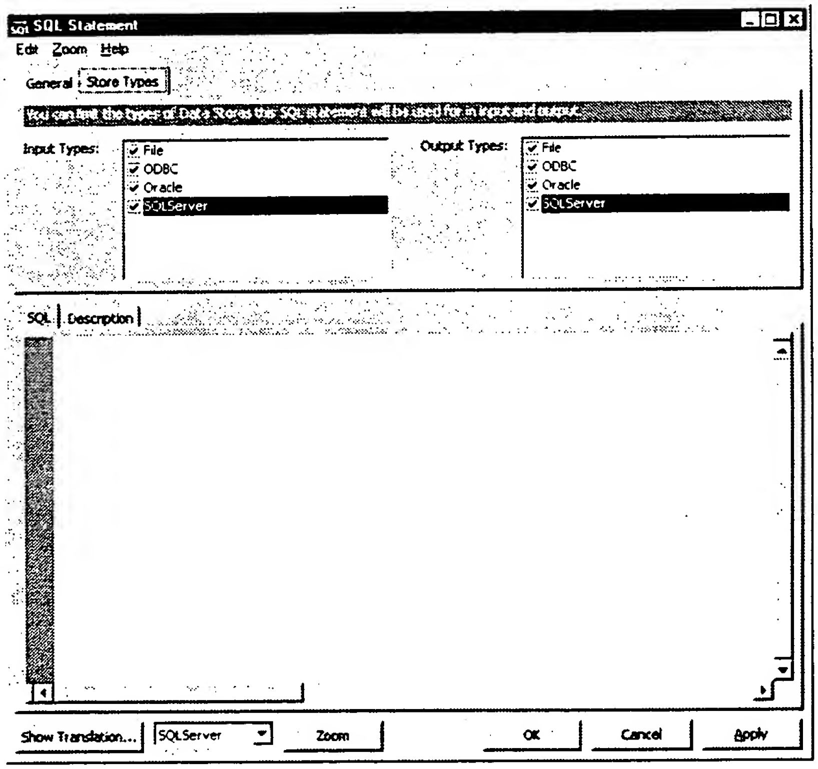
You may select multiple input types. For example, you could use the same SQL twice, once with SQL Server syntax and once with Oracle syntax, and have one or the other statements automatically disable itself when the source database is of the wrong type. This is of value when the group that contains the SQL statement is shared by multiple extractors.

Another approach to handling SQL dialect problems is to use E.piphany SQL replacement macros instead of database vendor-specific constructs. See Appendix A, "E.piphany Macros" for more information.

You can use the Store Types tab of the SQL Statement dialog box (Figure 61) to restrict data stores:

1. Check appropriate **Input Types**.
2. Check appropriate **Output Types**.

FIGURE 61: SQL STATEMENT DIALOG BOX: STORE TYPES TAB



You can limit the types of Data Stores this SQL statement will be used for in input and output.

Input Types: ☑ File ☑ ODBC ☑ Oracle ☑ SQLServer

Output Types: ☑ File ☑ ODBC ☑ Oracle ☑ SQLServer

SQLServer

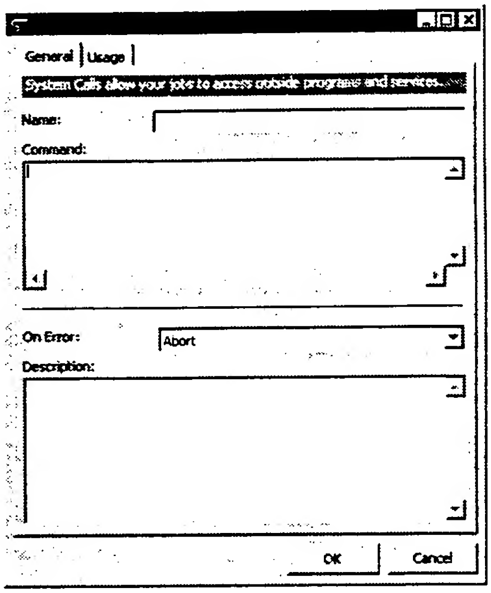
DEFINING SYSTEM CALLS

As mentioned, sites with more complex databases may require multi-stages and additional commands, such as lookup tables, aggregation splits, gathering data into ranges (binning), and duplicate detection. For this purpose, you may use system calls, which are executed during a job as if invoked from the console DOS command line.

To define a system call object:

1. Select the group folder for the system call object. The **System Call** folder is present by default although you can create system calls in any location.
2. Choose **System Call** from the **New** drop-down list.

FIGURE 62: SYSTEM CALL DIALOG BOX



System Calls allow your jobs to access outside programs and services...

Name:

Command:

On Error: Abort

Description:

3. Enter the name of the system call, the action to be taken upon error (abort, ignore, or print), the command line, and an optional description.

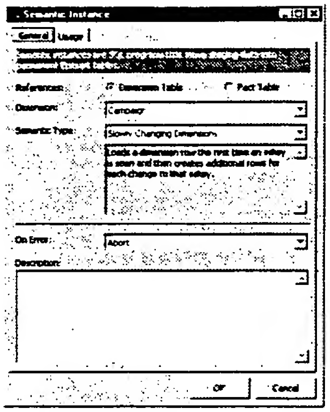
DEFINING SEMANTIC INSTANCES

To define a semantic instance object:

1. Select the group folder for the semantic instance.
2. Choose **Semantic** from the **New** button. (You can also right-click in the **Object Gallery** window and select **New Semantic** from the pop-up windows.) The Semantic Instance dialog box is displayed (Figure 63).
3. Select whether the semantic instance references a fact table or a base dimension table (the one to be operated on by the semantic type).

4. Select the associated table name.

FIGURE 63: SEMANTIC INSTANCE DIALOG BOX



[illegible]

References: ● Dimension Table ○ Fact Table

Dimension: Company

Semantic Type: Slowly Changing Dimensions

Loads a dimension row the first time an entity is seen and then creates additional rows for each change to that entity.

On Error: Abort

Description:

5. Select the semantic type from the drop-down list. See Appendix F, "Semantic Types" for descriptions of semantic types.
6. Select the action to be taken upon error: abort, ignore, or print.

DEFINING A TRUNCATION OBJECT

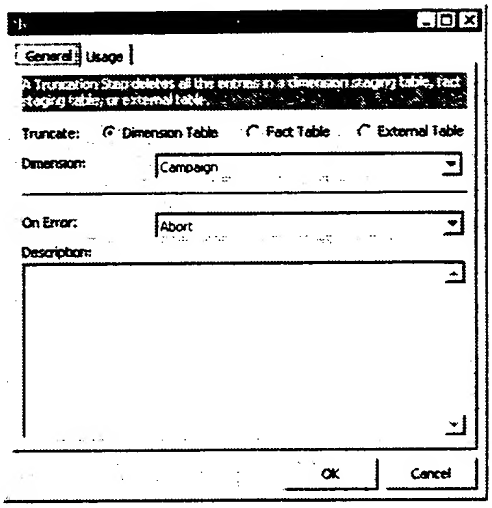
A truncation object can be added as a job step that truncates a specified table, either a dimension or fact staging table, or an external table. Typically, a job begins with the truncation (deletion) of old staging tables. Each table needs to be set up for truncation separately. You can create groups of tables to be truncated and reuse them in other jobs.

To define a truncation object:

1. Select the group folder for the truncation object.
2. Choose **Truncation** from the **New** menu in the Job Steps dialog box. The Truncate Steps dialog box is displayed (Figure 64).
3. Select the kind of table to be truncated (dimension, fact, external).
4. Select the name of the table to be truncated.

5. Select the action to be taken upon error: abort, ignore, or print.
6. Enter any description of this step.

FIGURE 64: TRUNCATE DIALOG BOX



A Truncation Step deletes all the entries in a dimension staging table, fact staging table, or external table.

Truncate: ● Dimension Table ○ Fact Table ○ External Table

Dimension: Campaign

On Error: Abort

Description:

DATA STORES TAB

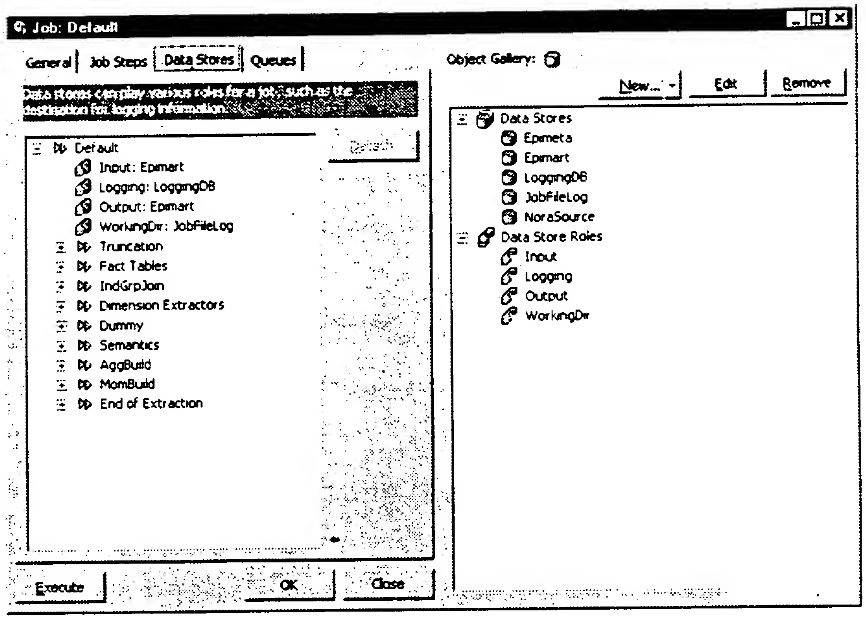
The Data Stores tab (Figure 65) of the Job dialog box allows you to assign the data stores and data store roles for a job and for job steps. The default data stores, data store roles, and any new data stores you have defined appear in the **Object Gallery**. For example, you can define custom data stores and data store roles and use them in system calls.

Assign a data store by selecting a data store object from the **Object Gallery** in the right window of the dialog box and dragging it onto its data store role in the left window. The object becomes attached. For example, select **Epimart** in the **Object Gallery** and drag it onto the **Output** role.

If a job step does not have a role assigned, it inherits the assignment from its first parent that does have a role assigned.

To remove an attached object, select it in the left window of the dialog box and click **Detach**.

FIGURE 65: JOB DIALOG BOX: DATA STORES TAB



Data stores can play various roles for a job, such as the destination for logging information.

Default
Input: Epmart
Logging: LoggingDB
Output: Epmart
WorkingDir: JobFileLog
Truncation
Fact Tables
IndGrpJoin
Dimension Extractors
Dummy
Semantics
AggBuild
MomBuild
End of Extraction

Data Stores
Epmeta
Epmart
LoggingDB
JobFileLog
NoraSource
Data Store Roles
Input
Logging
Output
WorkingDir

THE SCHEDULER

The Scheduler determines when queues are run. *Queues* consist of tasks, which are jobs or events tied to saved reports, such as chart-file generation and campaign-list pulls. A *campaign-list pull* refers to the Scheduler's finding out which campaigns are due for export and then generating export files and backfeed tables based on these campaigns.

The Scheduler runs as a Windows NT service (as does the AppServer). After a standard installation, the Services control panel has a listing for `instance_name_scheduler`, in addition to `instance_name` for the AppServer. Normally, both of these services are always running.

To set up the Scheduler, follow these steps:

1. Choose **Configuration** from the **EpiCenter** menu. In the General tab of the Configuration dialog box, note these two **config_master** entries that govern Scheduler invocation.
   - `queue_wakeup_interval` specifies how often the Scheduler service reloads the queues to check whether any tasks are due to start. A *task* may be a job within a queue or a Chart or Campaign queue scheduled to be run.
   - `queue_start_precision` specifies how close the task's start time should be to the current time in order for the task to start.
2. The default values are acceptable in most cases. If you need to modify them, select the key in the list and enter a new value in the **Value** text box.

   Increase these values only if you have many queues, and the time to process them (as evident from the log) approaches the value of `queue_wakeup_interval`.

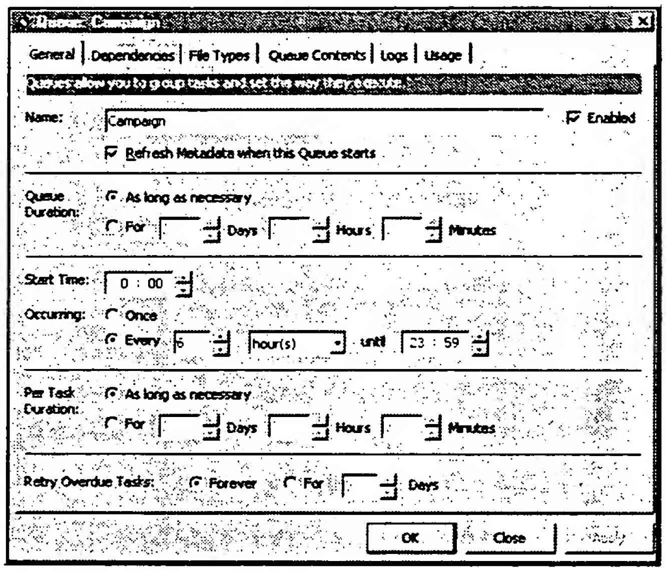
Note: *The value of `queue_start_precision` must be greater than half the value of `queue_wakeup_interval`. Otherwise, a queue could fail to run completely.*

3. Click **Update** to change the configuration parameters.

SETTING UP A QUEUE

Double-clicking the **Queues** folder in the EpiCenter Manager tree opens the Queue dialog box (Figure 66). This dialog box enables you to define a queue and view its contents.

FIGURE 66: QUEUE DIALOG BOX: GENERAL TAB



Queues allow you to group tasks and set the way they execute.

Name: Campaign ☑ Enabled

☑ Refresh Metadata when this Queue starts

Queue Duration: ● As long as necessary ○ For □ Days □ Hours □ Minutes

Start Time: 0 : 00

Occurring: ○ Once ● Every 6 hour(s) until 23 : 59

Per Task Duration: ● As long as necessary ○ For □ Days □ Hours □ Minutes

Retry Overdue Tasks: ● Forever ○ For □ Days

To set up a Queue, open the General tab of the Queue dialog box:

1. Enter the unique name of the queue.
2. Check **Enabled** in order for the queue to be run. All queues are disabled by default.
3. Indicate whether or not you want the metadata to be refreshed when the queue starts.

   Refresh metadata reloads all the metadata inside the Scheduler, similar to an AppServer refresh. The difference is that instead of the being associated with a Web page, a queue refresh is tied to a queue. Queues with Campaign and Chart file types need refreshing. For a job-only queue, in which metadata is loaded in a separate process, refresh metadata is an unnecessary overhead.
4. Indicate a duration of the queue: as long as necessary, or specify the number of days, hours, and minutes.

If **As long as necessary** is selected, the Scheduler does not terminate the queue. If a time is indicated, the queue runs this amount of time before the Scheduler service terminates it. This option is overruled, however, if a queue that can terminate the queue is scheduled to run.

5. Set the start time and indicate how often the queue should wake up: once, or every *x* number of hours until a specific time.

   All default queues wake up once per day (8:00 p.m). The maximum number of times that a queue can wake up is once each minute of the day, or 1440 times (24 x 60 = 1440).

6. Indicate how long to retry tasks that are overdue: either forever, or for a set number of days.

   If **Forever** is selected, whenever task execution is missed for any reason, the Scheduler attempts to execute the task the next time it wakes up the queue. When **For** is selected, the Scheduler re-executes the task only if its original execution date was no earlier than the specified number of days.

Note: *Reschedule your queues for Daylight Savings times so that they do not start during invalid HHMMSS combinations.*

You can use the Dependencies tab (Figure 67) to set up dependencies between this queue and other queues expressed in parent/child relationships. Parent queues must run before child queues. Child queues can run only after the parent queue finishes. If a child queue is already running, however, and the parent queue becomes ready to run, then the parent has to wait for the child to finish (unless the parent queue is set to terminate the child queue).

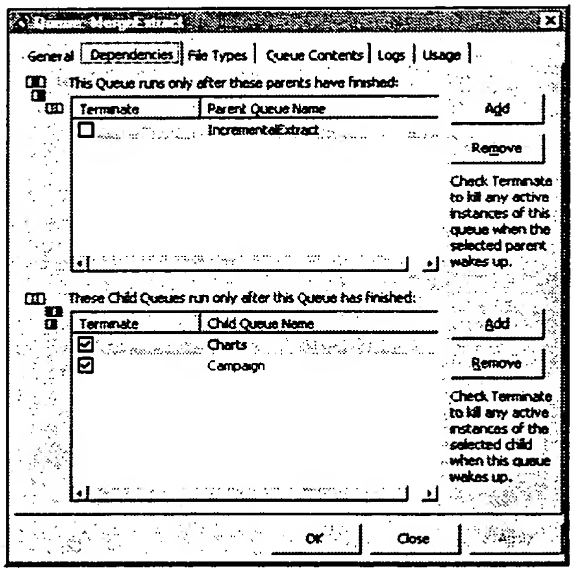
Note: *When making timing and dependency selections, estimate how long your queues might run and the optimal time for them to run. Also, consider whether queues might interfere with each other.*

In the upper pane, click **Add** and select any child queues that depend on the queue you are defining as a parent queue. If you check **Terminate** next to a child queue, any instance of this child queue that is running when the parent queue is ready to start is terminated. A parent queue cannot start if a child queue is running. It has to either wait for the child queue to finish, or terminate it.

In the lower pane, click **Add** and select any parent queues of which this queue is a child. Check **Terminate** to kill any instances of this queue that are running when the parent starts running.

Note: *Before modifying any dependencies, stop the Scheduler. After setting up dependencies between queues, restart the Scheduler.*

FIGURE 67: QUEUE DIALOG BOX: DEPENDENCIES TAB



This Queue runs only after these parents have finished:

Terminate	Parent Queue Name
□	IncrementalExtract

Check Terminate to kill any active instances of this queue when the selected parent wakes up.

These Child Queues run only after this Queue has finished:

Terminate	Child Queue Name
☑	Charts
☑	Campaign

Check Terminate to kill any active instances of the selected child when this queue wakes up.

To view the contents of this queue, click the Queue Contents tab (Figure 68). A task in the queue is shown by name, next schedule type (such as **As Soon As Possible**), next date (if appropriate), and the number of times that the task is expected to run during the day.

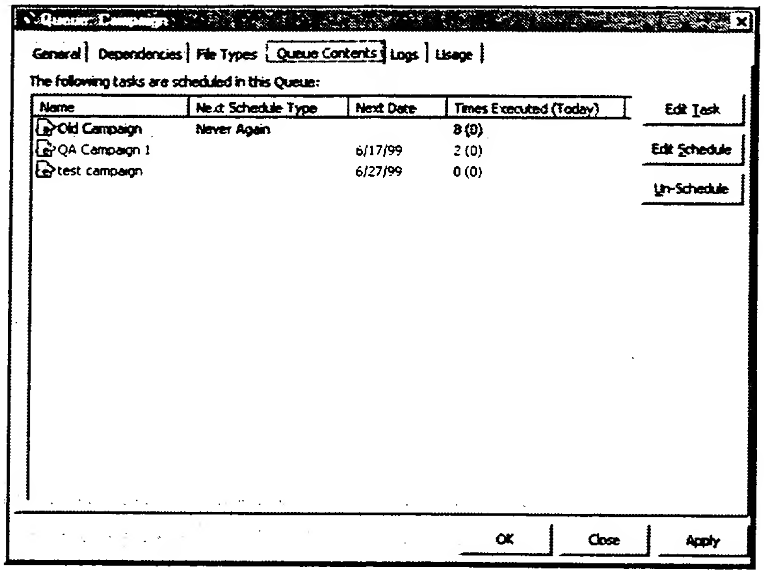
Note: ***As Soon As Possible*** *appears for newly added tasks. The first time that a a queue starts after a task has been added,* ***As Soon As Possible*** *is replaced by a value in the* ***Next Date*** *column.*

You can use this dialog box to remove a task from the queue by selecting it and clicking **Un-Schedule**.

To edit the queue dialog box for the job, click **Edit Task**.

To edit the schedule for this queue, click **Edit Schedule**, which opens the Task Schedule dialog box (Figure 70, on page 237).

FIGURE 68: QUEUE DIALOG BOX: QUEUE CONTENTS TAB



The following tasks are scheduled in this Queue:

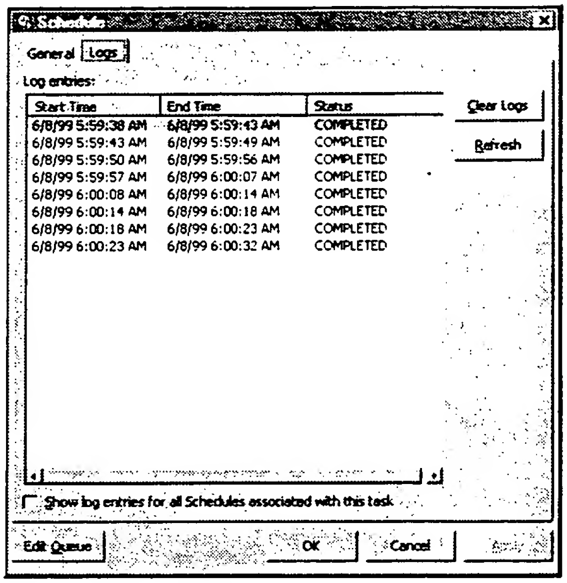
Name	Next Schedule Type	Next Date	Times Executed (Today)
Old Campaign	Never Again		8 (0)
QA Campaign 1		6/17/99	2 (0)
test campaign		6/27/99	0 (0)

The Logs tab of the Queue dialog box shows logs for the queue. A log entry contains the start and end times of the queue and its status: failed, completed, server aborted, and so forth. These logs report on whether the queue ran successfully, not whether members of the queue ran successfully. Therefore, a queue log can report that the queue ran successfully, although a task that ran as part of the queue failed.

To view the task logs, open the Queue Contents tab, select the instance of the task, and click **Edit Schedule**. In the Task Schedule dialog box (Figure 70, on page 237), click the Log tab to see the task logs.

The logs shown are for all instances of a task that have been run by this queue or that are about to be run by this queue (that is, all instances that are supposed to run as part of a currently running queue). Some scheduled tasks, such as jobs, can be in multiple queues. When **Show entries for all Schedules associated with this task** is unchecked, only instances associated with this queue are displayed. When the check box is checked, all instances, including ones associated with other queues, are displayed.

FIGURE 69: TASK SCHEDULE DIALOG BOX: LOGS TAB



Log entries:

Start Time	End Time	Status
6/8/99 5:59:38 AM	6/8/99 5:59:43 AM	COMPLETED
6/8/99 5:59:43 AM	6/8/99 5:59:49 AM	COMPLETED
6/8/99 5:59:50 AM	6/8/99 5:59:56 AM	COMPLETED
6/8/99 5:59:57 AM	6/8/99 6:00:07 AM	COMPLETED
6/8/99 6:00:08 AM	6/8/99 6:00:14 AM	COMPLETED
6/8/99 6:00:14 AM	6/8/99 6:00:18 AM	COMPLETED
6/8/99 6:00:18 AM	6/8/99 6:00:23 AM	COMPLETED
6/8/99 6:00:23 AM	6/8/99 6:00:32 AM	COMPLETED

□ Show log entries for all Schedules associated with this task

Clearing the logs for a queue does *not* reset the internal execution history. If a daily task has already run, and you clear the logs and re-set the daily run-time for later in the day, the task does not run again at the scheduled time, because it has already run for today.

SCHEDULING JOBS WITHIN QUEUES

Note: *Use the* **Queues** *folder to define queues. For instructions, see "Setting Up a Queue," on page 231.*

The Queues tab of the Job dialog box allows you to assign a job to one of the queues you defined in the **Queues** folder and to schedule this task in relation to other task in the queues. When you drag a job queue into the left window of this dialog box, the Task Schedule dialog box (Figure 70, on page 237) is displayed.)

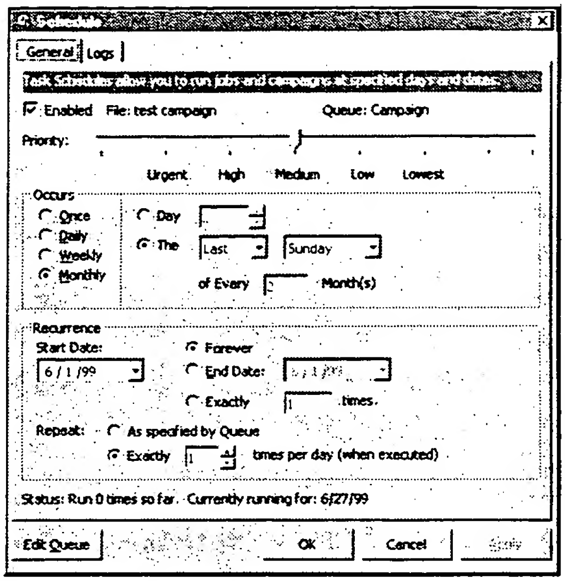
THE TASK SCHEDULE DIALOG BOX

The Task Schedule dialog box (Figure 70, on page 237) allows you to schedule tasks within queues (a *task* is any job, Campaign, or Chart). This dialog box is displays when you assign a job to a queue in the Job dialog box, and when you edit the schedule for any task. You can use the Task Schedule dialog box (Figure 70) to assign a priority and run-time parameters to tasks within a queue. The actual time that the Scheduler runs the queue runs depends on its relationship to other queues.

Note: *Preliminary values for each Campaign or Chart are assigned based on what the end user specified, but you can modify these.*

You can open a Queue dialog box for editing by clicking the **Edit Queue** button in the Task Schedule dialog box.

FIGURE 70: TASK SCHEDULE DIALOG BOX



Task Schedules allow you to run jobs and campaigns at specified days and dates.

☑ Enabled File: test campaign Queue: Campaign

Priority:

Occurs

○ Once
○ Daily
○ Weekly
● Monthly

○ Day
● The Last Sunday
of Every Month(s)

Recurrence

Start Date: 6/1/99

● Forever
○ End Date:
○ Exactly 1 times.

Repeat: ○ As specified by Queue
● Exactly 1 times per day (when executed)

Status: Run 0 times so far. Currently running for: 6/27/99

Follow these steps to schedule an task within a queue:

1. Assign this task a priority in relation to other tasks in the same queue. When you move the slider, the integer priority value, which ranges from 0 to 100, is displayed.
2. If **Enabled** is checked, the task can be executed.
3. Indicate how often the task occurs (once, daily, weekly, monthly).
4. Set starting and ending dates, or specify the number of times that the task should run. When you specify an exact number of times for it to run, the current number of times actually run is shown in this dialog box (after it has run at least once with this schedule).

THE CAMPAIGN AND CHART QUEUES

File types reflect the association of a report with a queue. The campaign file types in the default Campaign Queue are:

- GroupCampaignFile
- GroupCampaignListFile
- IndividualCampaignFile
- IndividualCampaignListFile

Whenever a campaigned is saved, the system places it into one of the system-defined file types associated with a Campaign queue. Similarly, the system assigns Chart file types to Chart queues. The default Chart queue has the file type, TicksheetFile.

Note: *Do not change the default file types in the File Types tab of Campaign- and Chart Queues dialog boxes. They are pre-configured.*

When the queue wakes up, it finds all saved reports of this type that were scheduled to run at least once more today, or that have not run as often as they should have on a previous day (but have not yet expired). All of the overdue, but unexpired, task instances are put into the queue, along with one instance of every task that needs to run today. So, for example, if a task is supposed to run four times a day, but it ran only once yesterday, and not at all today, then assuming that tasks expire after more than a day, when the queue wakes up, it plans to execute all three of yesterdays' missed task execution, plus one more for today.

Running a Campaign queue results in campaign export. Seeds are included in campaign output only after the Scheduler runs a Campaign queue.

EXTERNAL TABLES

Extraction statements are sometimes directed to load external tables that serve as intermediary tables for multi-staged extraction. E.piphany supplies one external table called `last_extract_date.` Use of this table is recommended, but optional. See "External Tables," on page 76.

To define external tables and their columns:

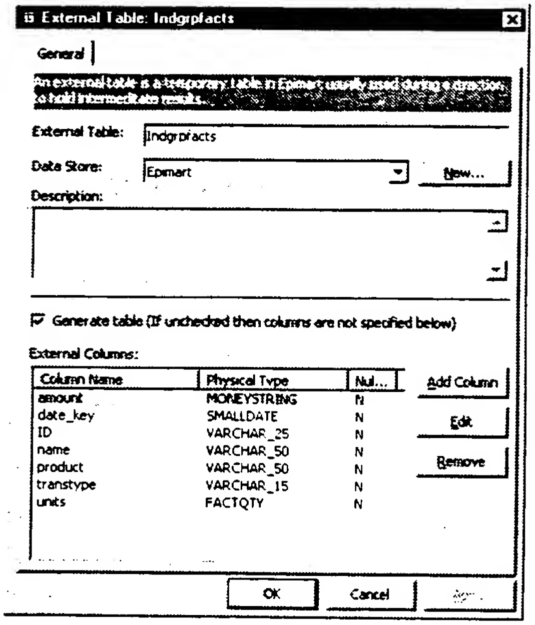
1. Right-click the **External Tables** folder and select **New External Table**.
2. Enter the name and any description for the table in the External Table dialog box (see Figure 71).
3. Select the Data Store for this table from the drop-down list.

   An external table must be in the `Epimart` datamart store in order to be generated.

   Click **New** to create a new data store as described in "The Data Store Dialog Box," on page 211.
4. By default, the **Generate** option is checked.

   Check **Generate** if you want to make the external table the target of an extraction statement and prefer to have the schema generator generate the table. This saves you from having to manually enter the column names. Unchecking this option means that you have to add the columns (as described below) to create this table.

FIGURE 71: EXTERNAL TABLE DIALOG BOX



An external table is a temporary table in Epimart usually used during a direction to hold intermediate results.

External Table: Indgrpfacts

Data Store: Epimart

Description:

☑ Generate table (If unchecked then columns are not specified below)

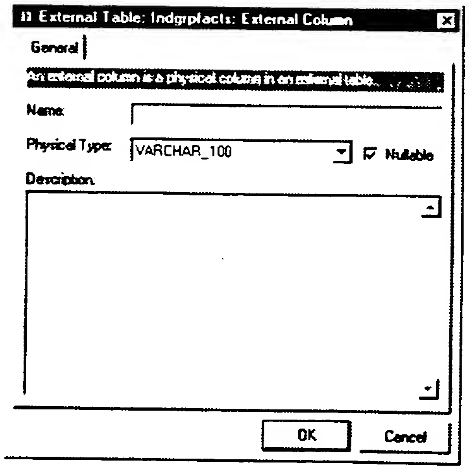
External Columns:

Column Name	Physical Type	Nul...
amount	MONEYSTRING	N
date_key	SMALLDATE	N
ID	VARCHAR_25	N
name	VARCHAR_50	N
product	VARCHAR_50	N
transtype	VARCHAR_15	N
units	FACTQTY	N

To add a column:

1. Click **Add Column** and enter the name in the External Column dialog box (Figure 72).

FIGURE 72: EXTERNAL COLUMN DIALOG BOX



An external column is a physical column in an external table.

Name:

Physical Type: VARCHAR_100 ☑ Nullable

Description:

2. Select the physical type. See Appendix D, “Physical Type Values” for descriptions of these physical types.
3. If **Nullable** is checked, then the external column allows null values.
4. Add as many columns as necessary. After you click **OK** in this dialog box, the columns are added to the **Column Name** list in the **External Columns** pane of the External Table dialog box.

MACROS

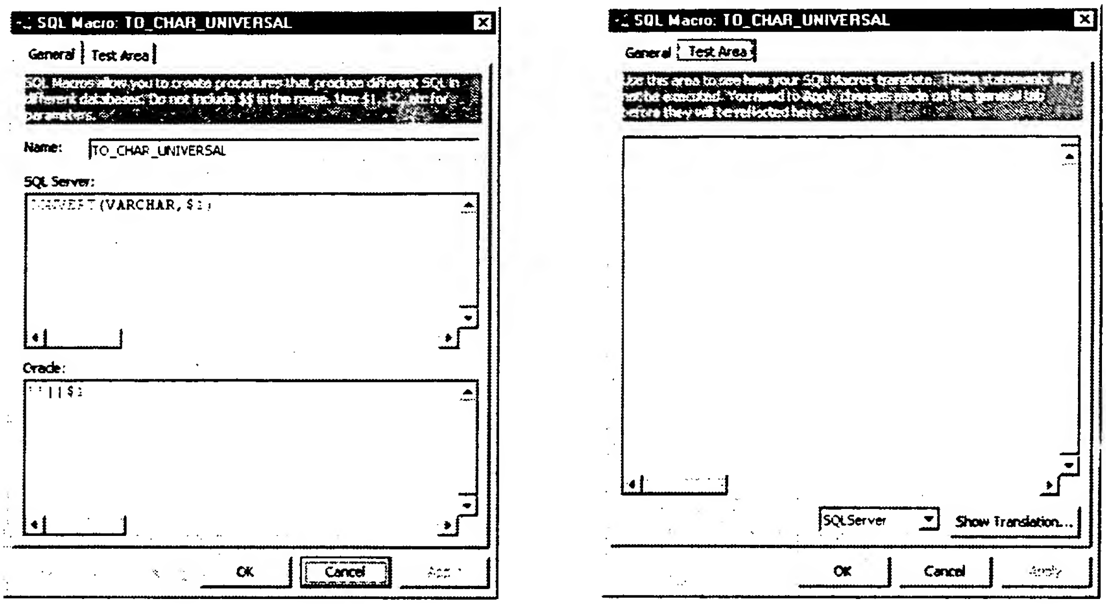
E.piphany provides optional SQL macros as described in Appendix A, “E.piphany Macros.” You may also use EpiCenter Manager to create your own SQL macros. Follow these steps:

1. Right-click the **Macros** folders and select **New Macro**.
2. In the General tab of the SQL Macro dialog box, enter the macro name without the symbol `$$`.
3. Enter the SQL syntax for both SQL Server and Oracle, unless you want the macro to translate to nothing in one database.

   Reasons for not translating a macro are that it not be necessary for one vendor (the `$$NO_FROM_LIST` macro on SQL Server), or it may be specific to a vendor (the `$$ORACLE` macro).
4. Click the Test Area tab to view the macro’s translation.
5. Select whether you want to display the translation for SQL Server or Oracle, and click **Show Translation**.

When viewing the translated SQL, note that any occurrences of the **$$** symbol are highlighted in red to make them easy to find. These macros did not translate because they do not exist, often because of a typo or other error.

FIGURE 73: SQL MACRO DIALOG BOX



SQL Macro: TO_CHAR_UNIVERSAL

SQL Macros allow you to create procedures that produce different SQL in different databases. Do not include $$ in the name. Use $1, $2, etc for parameters.

Name: TO_CHAR_UNIVERSAL

SQL Server:

```
CONVERT(VARCHAR, $1)
```

Oracle:

```
''||$1
```

SQL Macro: TO_CHAR_UNIVERSAL

Use this area to see how your SQL Macros translate. These statements will not be executed. You need to Apply changes made on the general tab before they will be reflected here.

SQLServer

LIST MANAGER EXTRACTION

To run List Manager, configure an appropriate default job in the **Jobs** folder. When List Manager is installed, the order of the extraction should be as follows:

1. SQL/Semantics
2. AggBuilder

Note: *Aggbuilder **must** run before MomentumBuilder in merge and rebuild jobs.*

3. MomentumBuilder
4. e4 End of Extraction

5. After finishing the extraction, refresh the AppServer as described in Chapter 5, "The E.piphany Application Server."

For information about running MomentumBuilder as a stand-alone executable and to determine if MomentumBuilder extracted the correct data, see "The MomentumBuilder Program," on page 111.

CONFIGURING E-MAIL

To have the system automatically send e-mail notification of the outcome of a job (either successful completion or job failure), EpiCenter Manager needs to know your e-mail Profile name and password. These must match the information in the **Mail and Fax or Mail** Control Panels. Windows uses this Control Panel to define the profile of mail accounts, where each profile could tie users to different mail servers. The operating system then makes mail to this server possible via the **SMAPI** program interface used by EpiChannel.

To determine your e-mail profile name, follow these steps:

1. Open the **Mail and Fax Control Panel** (from the **Start** menu, choose **Settings\Control Panel** and double-click **Mail and Fax**).

   You may be given a list of profiles or may be placed into the only profile available. If you do not remember your profile name, you can click **Show Profiles** to display the names of the existing profiles.

2. Choose **Configuration** from the **EpiCenter** menu.

   The list of EpiCenter configuration variables includes **Mail Password** and **Mail Profile Name.**

3. Click **Mail Profile Name** and enter the name of the profile as shown in the **Mail and Fax Control Panel** in the **Value** text box. Click **Update.**

4. Click **Mail Password** and enter the password. Click **Update.**

   The mail password depends upon which mail server is used and follows the rules of this server. Often, the password is the same as the Windows NT user password.

Note: *Because this password may be visible to others in mail logs or in metadata exports, do not use a password of any importance. Create a new mail user on your mail server exclusively for EpiChannel if revealing this password presents a problem.*

VERIFYING THAT E-MAIL NOTIFICATION WORKS

To test if e-mail is configured correctly, follow these steps:

1. Open a Job dialog box for a job that simply runs and exits. In the General tab, enter the addresses for mail on success and mail on failure.
2. Run the job.

If you receive e-mail notification, you have set up EpiChannel e-mail correctly. If not, repeat the steps for configuring e-mail given above and run another job. Please contact E.piphany Customer Support if there is still a problem.

CONFIGURING OUTLOOK EXCHANGE FOR EPICHANNEL

Follow these steps to configure Microsoft Outlook Exchange 98 for EpiChannel e-mail notification:

1. Install Microsoft Outlook Exchange 98 according to the instructions on the screen. When prompted to select which kind of installation, select **Corporate\Workgroup**. Reboot your computer.
2. Start Outlook Exchange and choose to configure an Internet Mail account.
3. When prompted, enter a **Profile Name** for this account.

Note: *The Mail Profile in the EpiCenter Manager Configuration dialog box is the default e-mail address for the EpiCenter Manager user.*

4. Set up the account with an appropriate e-mail account name, user name, and organization.
5. For the e-mail and reply address, enter a bogus address, such as *bogus@xyz.com*.

6. In the Servers tab, set both the incoming and outgoing mail servers to the name of the mail server that can route mail outside the company's firewall. (This name is not case-sensitive.) In the Incoming Server section, enter any account name and password. (Because EpiChannel never uses Outlook Exchange to check incoming mail, these can be any values.) Accept the default values for the other tabs.
7. Exit Outlook Exchange.

Note: *If the mail server is installed on a UNIX platform, you can test it by logging on with a shell account and sending mail to yourself and E.piphany from the UNIX mail program.*

Purging EpiMart Tables

You can remove tables from the EpiMart that are no longer needed. For example, you may have unused tables to delete or, as a result of metadata changes, have decreased the number of aggregates. These higher numbered aggregate tables remain in effect until you purge the EpiMart tables.

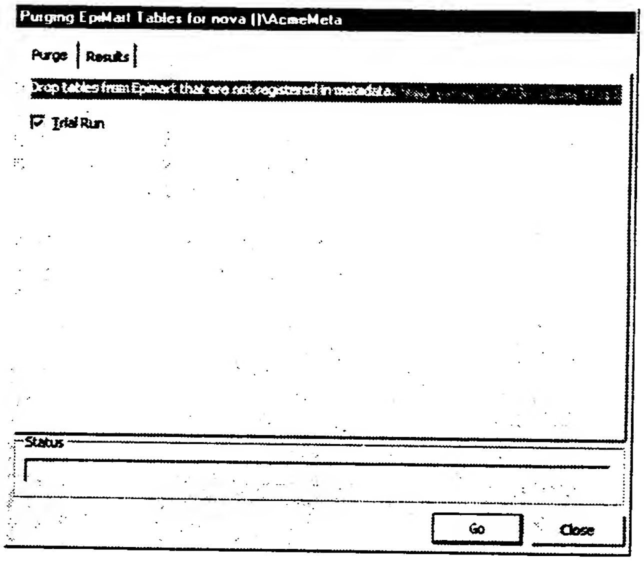
You should first try a trial run.

To purge database tables:

1. Choose **Purge Epimart Tables** from the **EpiCenter** menu.
2. Check **Trial Run** in the Purge tab, and click **Go**.

After the run, click the Results tab to view the tables to be deleted. If these results are acceptable, return to the Purge tab, uncheck **Trial Run**, and click **Go** to delete these tables from your EpiMart.

FIGURE 74: PURGE EPIMART TABLES DIALOG BOX



Purging EpiMart Tables for nova ()\AcmeMeta

Drop tables from Epimart that are not registered in metadata.

☑ Trial Run

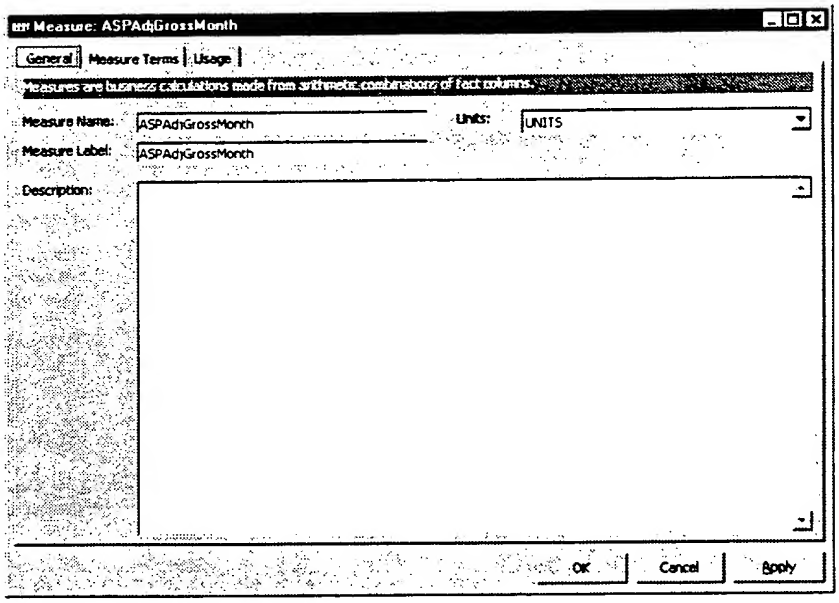
MEASURES

A *measure* is a business calculation that is the result of an arithmetic combination of fact columns. Each EpiCenter has associated measures, which are organized alphabetically by name in the EpiCenter Manager's **Measures** folder. At the frontend, a user selects measures on a Web page. The calculations that apply to this query depend on how you define the measures.

The **Measures** folder consists of the sub-folders: **Measures, Measure Layouts, Measure Sets,** and **Transaction Filter** sets. To define measures for the EpiCenter, follow these steps:

1. Right-click the **Measures** folder and select **New Measure** from the pop-up menu. The Measure dialog box (Figure 75) is displayed.

FIGURE 75: MEASURE DIALOG BOX: GENERAL TAB



Measures are business calculations made from arithmetic combinations of fact columns.

Measure Name: ASPAdjGrossMonth

Units: UNITS

Measure Label: ASPAdjGrossMonth

Description:

2. In the General tab, enter a name for the new measure.
3. If the label (the name visible to end users) differs from the name, replace the label name.
4. Enter a description of this measure (for your reference only).

5. Select the unit of measurement for the calculations from the **Units** drop-down list. A default EpiCenter configuration has these measure units:

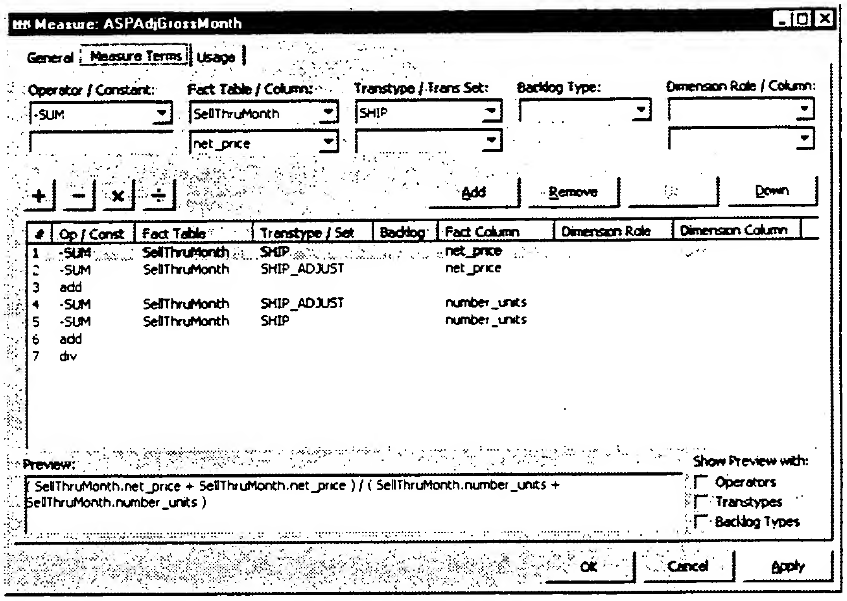
    - **Currency_Local** (for local monetary units, such the Franc or Yen).
    - **Currency_US** (for currency expressed in U.S. dollars)
    - **Percent** (for percentages)
    - **Units** (for the count of an item)

    You can define unit options via the Measure Units tab of the EpiCenter Manager's Configuration dialog box (see "Measure Units," on page 420).

MEASURE TERMS

You can use the Measure Terms tab of the Measure dialog box (see Figure 76) to define how the measure is calculated. A *measure term* is one component of an arithmetic expression that makes up a measure. A measure term refers to the aggregation of a single fact column in a fact table, such as `SUM(Order.net_price)` with a particular transaction type (or transaction type set). Each term applies to a specific column in a fact table, transaction type or set, and backlog type (if applicable).

FIGURE 76: MEASURE DIALOG BOX: MEASURE TERMS TAB



Operator / Constant: -SUM

Fact Table / Column: SellThruMonth / net_price

Transtype / Trans Set: SHIP

Backlog Type:

Dimension Role / Column:

#	Op / Const	Fact Table	Transtype / Set	Backlog	Fact Column	Dimension Role	Dimension Column
1	-SUM	SellThruMonth	SHIP		net_price		
2	-SUM	SellThruMonth	SHIP_ADJUST		net_price		
3	add						
4	-SUM	SellThruMonth	SHIP_ADJUST		number_units		
5	-SUM	SellThruMonth	SHIP		number_units		
6	add						
7	div						

Preview:

(SellThruMonth.net_price + SellThruMonth.net_price) / (SellThruMonth.number_units + SellThruMonth.number_units)

Show Preview with:
□ Operators
□ Transtypes
□ Backlog Types

One measure term is combined with other measure terms to create a composite measure. Each defined measure term is a numbered step that displays in the lower pane of the Measures Terms tab. The E.piphany system converts these steps to appropriate SQL SELECT statements.

Note: *Reverse Polish Notation is the mathematical notation used to construct measure definitions. See "Reverse Polish Notation," on page 252 for more information.*

Use the upper pane of the Measure Terms tab to define a step for a measure term. After defining the term, click **Add** to add it as the next step in the lower pane:

1. Select an operator.

   Either select one of the SQL operators from the drop-down list: **SUM**, **MIN** (minimum values on a fact column), **MAX** (maximum values on a fact column), **COUNT**, **COUNT DISTINCT** (or the negative values of these operators, such as **-SUM** or **-COUNT DISTINCT**), or enter a constant character value in the text box.

When counting operations are performed on dimension tables, the effect of slowly changing dimensions can result in duplicate counts. To avoid this, use a measure that performs a `COUNT DISTINCT` on the fact table with reference to the unique ID column of the dimension role (assigned when you created the demographic base dimension table). Otherwise, in a slowly changing dimension, the `COUNT DISTINCT` operator would count a single element each time that it appears in the dimension table.

2. Select the fact table and column where the data resides.
3. Select an associated transaction type, or one of the transaction type sets defined as part of your schema.

   All facts have an associated transaction type. For a description of the kinds of transaction types, see Appendix B, "EpiCenter Configuration."

If you select a Transaction Type Set, a single measure term calculation (usually `SUM`) can be used to add up all rows for all transaction types in the set.

4. Select the backlog type, if appropriate.

   The backlog types are **BEGIN** or **END**; or leave blank if the backlog type does not apply. You can use a backlog type when a measure term (a single line of the measure definition) should exhibit accumulating behavior. Normally, when running a report of Sales by Month, for example, the report shows only the sum of transactions that occurs in each month of the report.

   When a backlog type is used, however, the columns of the report show accumulated values from previous months, in addition to the current month. You can use **BEGIN** to show the accumulated value at the beginning of each period, and **END** to show the ending accumulated value.

   For example, assume that report of sales by month transactions shows $10 for June, $20 for July, and $40 for August. A report of the beginning backlog shows the accumulated value at the beginning of each month:

June	July	August	September
$0	$10	$30	$70

A report of the ending backlog shows the accumulated value at the end of the month:

June	July	August
$10	$30	$70

Note: *Backlogs are always calculated based on the built-in date dimension role. User-defined dimension roles that refer to the date dimension are not available for backlog calculations.*

5. For **COUNT DISTINCT** operators only, select the associated dimension role and column (which should have a unique ID column).

   This enables the counting of the distinct values for a specific column in a dimension table related to Campaign Manager or List Manager.

6. Click **Add** to add this as the first step in the **Measure Terms** pane.
7. If appropriate, add another measure term by repeating the above steps.
8. Click the appropriate arithmetic operator to be applied to the measure terms: add (**+**), subtract (**-**), multiply (**x**), or divide (**/**). See "Reverse Polish Notation," on page 252 for examples.

The **Preview** pane of the Measure dialog box (see Figure 80) shows the translation of the Reverse Polish Notation calculations to standard (infix) calculations. If the measure calculation does not currently make sense (for example, a term is followed by two adds), the notation in the lower pane displays in red with one or more missing or extra term tags. You can save the measure, but the AppServer cannot use it.

You can choose to display only operators, transtypes, or backlog types, or any combination of these in the Preview.

Warning: *The removal of fact or dimension columns from the EpiCenter can remove measure terms, which invalidates their calculations. In this case, check all of the measures to determine if there are any Preview displays that appear as red.*

To edit an existing measure, double-click its **Measure** folder, which displays the Measure dialog box. Modify this as described above. Click **OK** to save your changes.

To delete a measure, right-click its folder and select **Delete** from the pop-up menu. You can also use this pop-up menu to duplicate or export the measure.

CONFIGURING MEASURES FOR SCORING

You can use measures directly to score a list using the Scoring Web page. When configuring your EpiCenter, consider the kinds of measures you need to define for use with scoring. Some commonly used measures for RFM (Recency, Frequency, Monetary) analysis are as follows:

- The length of time since a customer purchased an item.
- The number of times a customer purchased the item.
- The total amount of money a customer has spent.

REVERSE POLISH NOTATION

The **Measure Terms** pane in the Measure dialog box uses Reverse Polish Notation (RPN) to construct measure definitions. (Reverse Polish Notation is named for its Polish inventor, Jan Lukasiewicz.) RPN operations are performed in a last-in, first-out (LIFO) basis. All of the values to be operated upon are placed in a stack. Then the top two are operated upon and the result of that operation is placed in the stack, replacing the previous two values. Then the next top two are operated on and the result placed in the stack, and so forth.

For example, using Reverse Polish Notation for this calculation:

```
1 + (2 * 3) = 1, 2, 3, *, +
```

means that 1, 2, and 3 are placed in the stack. The last two items (3 and 2) are multiplied, and the resulting value 6 is placed in the stack. The stack now holds 6 and 1, which are added, and the result 7 is placed in the stack.

In contrast, applying RPN to the calculation:

```
(1 + 2) * 3 = 1, 2, +, 3, * in RPN
```

means that 1 and 2 are placed in the stack. These items are added, and the result 3 replaces them. Next, the value 3 is placed in the stack (the stack now holds 3 and 3). These items are multiplied and the result is 9.

The following example shows how RPN is used to define a measure definition (in the Measure dialog box). The Average Sales Price (ASP) for Booked/Gross Orders equals the total number of dollars received, divided by the total number of units shipped.

```
SUM (Order.net_price)
SUM (Order.number_units)
div
```

This example is calculated using RPN as follows. The sum of all of the Order net prices is calculated and placed in the stack. Then the sum of all of the number of units is calculated and placed in the stack. Next, the division operator is applied to the top two items in the stack. The result equals the ASP.

For the operators that apply to the aggregates, use the arithmetic operators: add, sub, mult, and div.

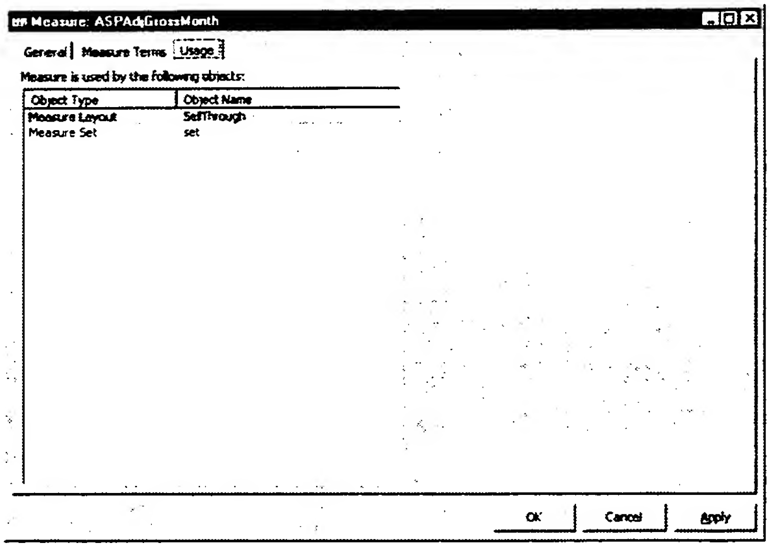
The next example a similar calculation. This time the ASP is calculated for booked net orders (booked orders minus returns). Here the sums of two Order net prices are added (the returned items are represented as a negative number) and placed in the stack. Then the sums of two Order number of units (the returned items are represented as a negative number) are added and placed in the stack.

```
SUM (Order.net_price) BOOK
SUM (Order.net_price) BOOK_RETURN
add
SUM (Order.number_units) BOOK
SUM (Order.number_units) BOOK_RETURN
add
div
```

DEFINING MEASURE USAGE

To view the objects that use the measure, click the Measure Usage tab of the Measure dialog box (Figure 77). This dialog box tells you what other objects use this Measure object. (Editing or removing a global object affects all usages of the object.)

FIGURE 77: MEASURE DIALOG BOX: USAGE TAB



Measure is used by the following objects:

Object Type	Object Name
Measure Layout	SellThrough
Measure Set	set

MEASURE LAYOUT

Defining the measure layout for a Web page is a two-step process. First, you can add all of the measure column elements that you want to appear on the final Web page and arrange them by column. This sets up the organization of the Web page and assigns names to the measure column elements in the Web page (the actual value is derived from the measure to which you map this combination of elements).

Second, you need to map each set of measure column elements that a user might choose from the Web page to the measure calculation that this choice represents. (Measure layouts apply to all reporting and analysis Web page types except Influences and Cluster, which use measure sets instead.)

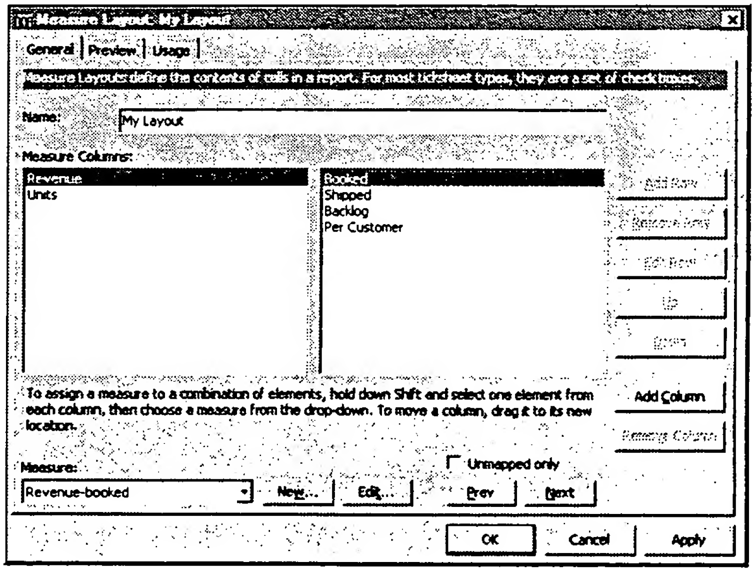
You can create a new measure within the Measure Layout dialog box. Click **New Measure**, which opens the Measure dialog box (see Figure 75). Follow the instructions given in "Measures," on page 246.

ADDING ELEMENTS TO COLUMNS

To add elements in the appropriate columns:

1. Open the General tab (Figure 78) in the Measure Layout dialog box. Note that a single empty column is displayed for a new measure layout.
2. Click **Add Row** to define the measure column elements for this column.

FIGURE 78: MEASURE LAYOUT DIALOG BOX: GENERAL TAB



Measure Layouts define the contents of cells in a report. For most tickshset types, they are a set of check boxes.

Name: My Layout

Measure Columns:

Revenue	Booked
Units	Shipped
	Backlog
	Per Customer

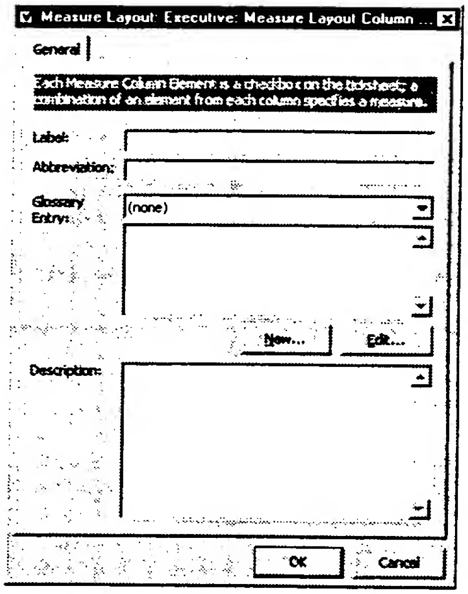
To assign a measure to a combination of elements, hold down Shift and select one element from each column, then choose a measure from the drop-down. To move a column, drag it to its new location.

Measure: Revenue-booked

□ Unmapped only

3. In the Measure Layout Column Element dialog box (Figure 79), enter the label (the name as it appears in the Web-page column), an abbreviation (if appropriate), and a description for your reference.

FIGURE 79: MEASURE LAYOUT COLUMN ELEMENT DIALOG BOX



Measure Layout: Executive: Measure Layout Column ...

Each Measure Column Element is a checkbox on the tickSheet; a combination of an element from each column specifies a measure.

Glossary Entry: (none)

4. To set up a glossary entry, click **New** and type the entry name and help text (that displays to end users) in the Glossary dialog box (see Figure 82).

Note: *The description you enter in the Help Text box can include HTML coding for links. For an example, see "Glossary Entries," on page 262.*

By defining glossary entries, measure column elements in Web pages, such as Units, Gross, and Sell-Through, are hyperlinked to a glossary page. When a user clicks a link, a glossary page displays which defines it. These glossary entries help users to understand your terminology.

5. Click **OK** to add the measure column element to this column.
6. Click the **Add** button again to add another measure column element to the column. Continue to add elements as described above until you have entered all of the elements for the column.
7. Click the **Add Column** button to add a second empty column. Click **Add** to enter the new column's measure column elements as described above.

8. Click **Add Column** to add another empty column, if applicable. There is a maximum of five columns per Measure Layout. Add the measure column elements to the new column.
9. To change the order of one column with another (the order in which they display on the Web page), select a column and drag it onto the new column location. The columns' contents are switched.
10. To remove a column, select it and click the **Remove Column** button.

Mapping Elements to a Measure

End users can select any combination of measure column elements (one per column) on a Web page. Each combination of elements equals a measure (whose calculations determine the contents of the generated report or query). For each Web page, you need to map every measure column element in a given column to every other measure column element in all of the other columns (to cover all of the possible combinations that a user may select).

To map elements to a measure:

1. While holding down the **Shift** key, select an element from each column. Each selection is highlighted.
2. With all of your selections highlighted, choose a measure from the drop-down list.

This measure is invoked by the system whenever the user selects this combination of elements on the Web page.

Verifying that All Elements are Mapped

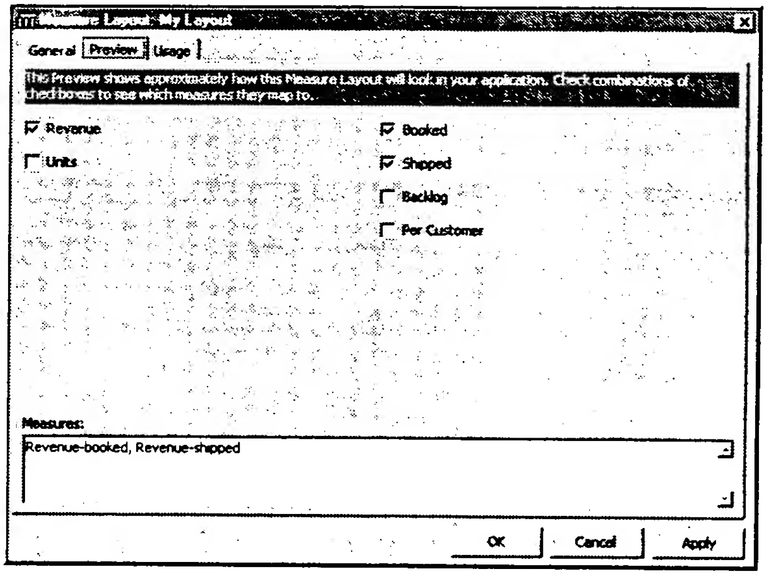
Click the **Previous** and **Next** buttons to cycle through the combinations of measure column elements that comprise a measure. The mapped element in each column is highlighted, and the measure that these elements map to is shown in the Measure text box.

Check **Unmapped Only** to display only those combination of elements that have *not* been mapped. (Remember that each distinct path through the columns must be mapped.)

Click the **Previous** and **Next** buttons to cycle through the combination of elements. Unmapped elements are highlighted, and no measure name is displayed in the Measure text box.

You can use the Measure Layout Preview tab to view how the element layout appears on the Web page. Make one or more selections per column; the associated measure or measures are shown in the Measures text area.

FIGURE 80: MEASURE LAYOUT DIALOG BOX: PREVIEW TAB



This Preview shows approximately how this Measure Layout will look in your application. Check combinations of checkboxes to see which measures they map to.

☑ Revenue
☐ Units

☑ Booked
☑ Shipped
☐ Backlog
☐ Per Customer

Measures:
Revenue-booked, Revenue-shipped

MEASURE SETS

Instead of measures, Influences and Community Clusters Web pages use measure sets (one or more measures that work in combination as a set). There are three types of measure sets: Classification, Regression, and Clustering. Influences Web pages use the Classification and Regression types. Community Clusters Web pages use the Clustering types.

To answer end-user queries, the E.piphany system builds models that use two types of trees: classification and regression. A *classification tree* finds rules that can predict the value of a discretely valued *attribute* based on the values of a set of other attributes. A *regression tree* finds rules that can predict the value of a particular numeric *measure*, such as customer profitability, based on the values of a set of attributes, such as customer attributes. The difference between classification and regression is that classification trees use attributes to predict an attribute, and regression trees use attributes to predict a measure. The basic questions answered by the two types of trees are very similar, but they require slightly different configuration.

Clustering is the process of finding groupings in data. The E.piphany system identifies groups as places of high concentration of data points. These groups are usually defined by some kind of internal consistency, such as households with similar demographics, or individuals with similar buying patterns. Community Clustering is similar to Influences, but does not have a target variable.

Measure set types have associated roles, which may be either **Count**, **TargetSum**, or **SumSquared**. All measure sets contain a **Count** role measure. (Classification and Clustering measure sets must contain only the **Count** measure role.) Measure sets of the Regression type must contain the **Count** and **TargetSum** role measure, and optionally, the **SumSquared** role measure.

The **Count** role is a measure that counts the number of rows in the primary dimension associated with the measure set, such as the count of the rows in the individual dimension. The **TargetSum** role is a measure that is the sum of the quantity you are trying to predict when using this measure set. For example, if you want to predict the total amount that customers purchased, the **TargetSum** role is a measure that is the sum of the purchases made by customers. Thus, if the value you want to predict is `X`, then the **TargetSum** role is a measure whose value is `SUM (X)`.

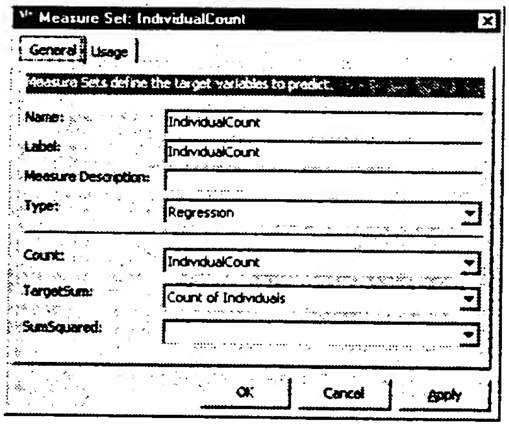
When the E.piphany system builds models that need to compute statistical variance, it computes the sum of the squares of the value you are trying to predict for customers. For example, if the target value you are trying to predict is `X`, then the **SumSquared** role is a measure whose value is `SUM (X * X)`. Instructions for specifying the optional **SumSquared** role are given below.

A Classification measure set is always associated with an attribute. If that attribute is a List Membership attribute, then lists are used with the measure set. Attributes with dimension roles and dimension columns (regular attributes) specify the target attribute. Attributes are not associated with regression and clustering measure sets.

To define a measure set for classification:

1. Right-click the **Measure Sets** folder and select **New Measure Set.**

FIGURE 81: MEASURE SET DIALOG BOX



Measure Set: IndividualCount

Measure Sets define the target variables to predict.

Name: IndividualCount

Label: IndividualCount

Measure Description:

Type: Regression

Count: IndividualCount

TargetSum: Count of Individuals

SumSquared:

2. In the Measure Sets dialog box (Figure 81), enter the name of the measure set.
3. Enter its label (this is the name that end users see on Web pages).
4. When defining a measure set for a list membership target (that is, a measure set with no attribute and only the Count role defined), you can enter a short description in the **Measure Description** text box that shows up in the Influences Web page target selection box. This is useful for cases in which the list membership target is weighted by a measure that is not the count. For example, if the user associates Revenue with the Count role, then he or she may want to enter "weighted by revenue" in the description box of the measure set. Then, the target shows up in the Influences Web page as Member of List, weighted by revenue.

5. Select **Classification** as the measure set type.
6. In the Count drop-down list, select a measure that gives a count of the number of rows in the primary dimension. In some cases, you may wish to weight the rows in the primary dimension (for example, based on the amount of revenue from a customer). In such cases, the **Count** role should be assigned a measure such as a simple count of the units shipped, or a measure that sums the price of all products shipped, which produces a count that is weighted by price.

To define a measure set for the Community Clusters Web page:

- Select Clustering as the measure set type.
- Specify a **Count** measure role, which can be the count of the number of rows in the primary dimension, or possibly a weighted count (as described above for Classification measure sets).

Defining a measure set for regression is similar to defining one for classification:

- Select a measure for the **Count** measure role that counts the number of members of the primary dimension.
- Select a measure for the **TargetSum** measure role that is the sum of the fact column that corresponds to the fact value that you are trying to predict.
- In some cases, you may need to select a measure for the **SumSquared** measure role.

  Usually, **SumSquared** can be computed from the **TargetSum** role, since the sum value corresponding to each row in the primary dimension can simply be squared. If the rows of the primary dimension table with which the measure set is used represent actual individual or households, you do not need to specify the **SumSquared** role. The **TargetSum** role computes the total amount of the target measure for each individual or household. That is, it computes the value x for each individual or customer, and this value can be squared for each individual or household. The E.piphany system can square this value and then sum it to obtain the right result.

If the rows in your primary dimension table represent some aggregate of individuals or households (for example, customer types or household types), then specify the **SumSquared** role. The TargetSum *measure* contains the sum of the target value for each customer type. The **TargetSum** *role* gives you the target value for each customer *type* (a sum over the individual customers), not the target value for each customer. Squaring the **TargetSum** role for the customer type does not square the values of the individuals aggregated to produce the customer type. Instead, it squares the value for the customer type aggregate, which is incorrect.

In such cases, at extraction time, you need to set up a fact table that contains the squares of the target value for each individual customer, and then sum these squared values to produce the sum-squared values for the aggregate customer types. You can then use this fact table to define a measure for the **SumSquared** role that, for each customer type, gives the sum of the squares of the individual customer target values.

PRESENTATION

The **Presentation** folder includes sub-folders for **Glossary Entries**, **Attributes**, **Web Pages**, and **Topics**.

GLOSSARY ENTRIES

Glossary entries define elements on a Web page to end users. To set up a glossary entry:

1. Double-click the **Glossary Entries** folder, which displays the Glossary dialog box (Figure 82).
2. Enter the name of the glossary term.

3. In the **Help Text** box, enter the definition that you want end users to see. This description can include HTML coding for links. For example: This is the dollar amount of revenue recognized. For more details, see our policy at:

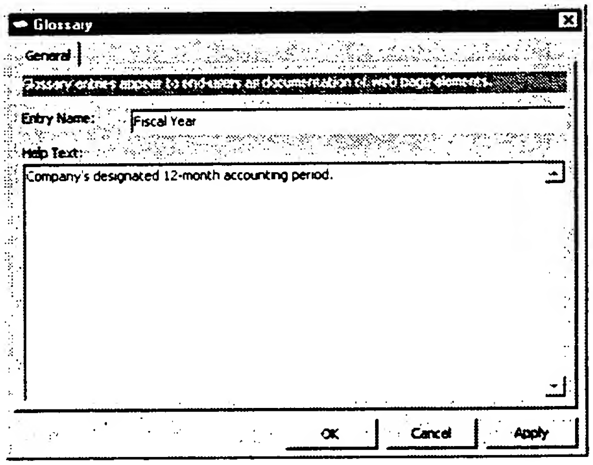
   ```
   <a href="www.company.com/policies/revrec.html">
   revenue recognition</a>
   ```

   The entry is added to the **Glossary Entries** folder.

4. To create additional entries, right-click the **Glossary Entries** folder, and select **New Entry**.

You can also define a new glossary entry within the Measure Layout and Attribute dialog boxes.

FIGURE 82: GLOSSARY DIALOG BOX



Glossary entries appear to end-users as documentation of web page elements.

Entry Name: Fiscal Year

Help Text:

Company's designated 12-month accounting period.

ATTRIBUTES

Attributes, which are derived from columns in dimension tables, appear as items that end users can select on Web pages. Attributes also serve as filters that users apply to results to refine (drill down) their queries. The only difference between an attribute in a table and one in the E.piphany system is that an E.piphany attribute has an associated display label, such as Fiscal Year, that can be configured via EpiCenter Manager. Within an EpiCenter, an attribute is a global object; however, only attributes (and attribute roles) appropriate to a specific dialog box are displayed.

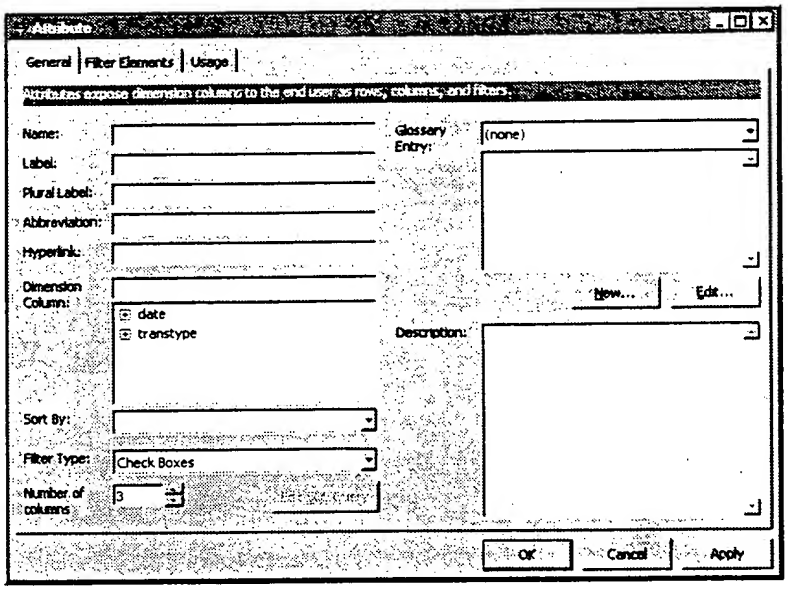
Note: *The **date_attributes.mdb** export file in the **ConfigFiles** directory contains common attributes based on the date dimension. If you import this file, you do not need to create attributes for fiscal year, month, and year name, and so forth. The filters for these attributes are dynamic–the check boxes/list boxes are based on the contents of your date dimension.*

To define an attribute, follow these steps:

1. Right-click the **Attributes** folder and select **New Attribute**.
2. In the Attribute dialog box (Figure 83), enter the name of the attribute and its label (the name that appears on the Web page).
3. Verify that the **Label Plural** is the correct plural of the label name (for multiple attributes).

4. Enter an abbreviation that the system can apply should the label name be too long (for example, **ASP** for Average Sales Price).

FIGURE 83: ATTRIBUTE DIALOG BOX: GENERAL TAB



Attributes expose dimension columns to the end user as rows, columns, and filters.

Glossary Entry: (none)

Filter Type: Check Boxes

Number of columns: 3

5. Enter an optional hyperlink format in the Hyperlink text box. When you do, each attribute value that appears in a Web pages is displayed as a live hypertext link. You can use the string `$val$` to indicate where the attribute value is to appear in the URL for the link; for example:

```
www.company.com/data/$val$
```

6. Select the dimension column for this attribute from the base dimension table pane. Click a plus sign to expand the tree.
7. If you have an alternate column within the dimension role that provides an appropriate sorting order, you can specify that column in the **Sort By** drop-down list. For example, assume that your attribute is a date column that stores dates as text strings of the form:

```
Month dy, year
```

In this case, it might make sense to display the attributes in calendar order rather than by month. Choose **(default)** for the native sorting order of the attribute itself.

8. Select the attribute's filter type from the **Filter Type** drop-down list. Then use the **Select Attribute Values** pop-up menu to choose its display format:

   - **Check Boxes**

     Check boxes are rectangles next to an item that a user points and clicks to check or uncheck. Users may check more than one item in a column.

     For the display format, select the number of columns for the attribute values from **Number of columns.**

   - **Dynamic Check Boxes**

     Dynamic check boxes are refreshed each time the AppServer starts up; static check boxes never change. For the display format, select the number of columns.

Note: *To generate check boxes and dynamic check boxes, you need to enter the proper SQL. Clicking the **Edit SQL Query** button in the General tab of the Attribute dialog box displays a template that you can modify for this SQL.*

   - **Dynamic Listbox**

     A dynamic list box contains a list of items for filter selection that are refreshed each time the AppServer starts up. The end user clicks the down arrow to the left of the box to display a drop-list menu of all list items.

     For the display format, select the height of this drop-down list from **List box height.**

   - **Entire Dimension**

     All attributes of the dimension are available as a filter. (This filter type is for special purposes in some data-mining algorithms.)

   - **List Membership**

     This drop-down contains a list of lists. These lists are created by end users and stored in the Report Gallery.

     For the display format, select the height of the drop-down list.

- **Radio Buttons**

  Radio buttons allow the end user to make one choice only. For the display format, select the number of columns.

- **Listbox**

  A list box contains a list of items for selection. The end user clicks the down arrow to the left of the box to display a drop-down of the items in the list.

  For the display format, select the appropriate height for this drop-down list.

- **Text Box**

  A text box is a blank area in which the user may enter text.

  For the display format, select the height of the text box from the counter labeled **Text box height**.

Note: The primary dimension attribute for an Influences Web page must have the filter type of Entire Dimension or List Membership, which is available for `indiv` or `group` only.

9. As with measurements, attributes can be defined in the glossary. You can select an existing glossary entry from the drop-down list, or create a new one.

   To set up a glossary entry, click **New** and type the entry name and help text (that displays to end users) in the Glossary dialog box (see Figure 82). The **Help Text** description you enter can include HTML coding for links.

FILTER ELEMENTS AND FILTER GROUPS

Filters allow Web page users to restrict the data accessed for a query to specific attribute values (values in dimension columns); for example, the data can be filtered so only the values for direct sales during the years 1990 through 1995 are returned. Filter elements are specific dimension-column values that users can choose. *A filter element* appears on the Filter Pop-up as a single check box or entry within a list box. A *filter group* is a logical grouping of filter elements. For example, a filter by year might have the a filter group of Q197 (first quarter of 1997), that includes the months January '97, February '97, and March '97.

Filter elements can be assigned to groups dynamically or statically. For dimension columns that contain changing values, it is often better to choose a dynamic filter type. For columns that seldom change, a static filter type (one for which the label in the General tab does not include the word "dynamic") is often preferred.

DYNAMIC FILTER GROUPS

When you choose a dynamic filter type, **Edit SQL Query** on the General tab is activated. Click this button to display the Fill from Query dialog box, which you can use to prepare an SQL query to populate the filter groups for this attribute. The AppServer invokes this query whenever it starts up.

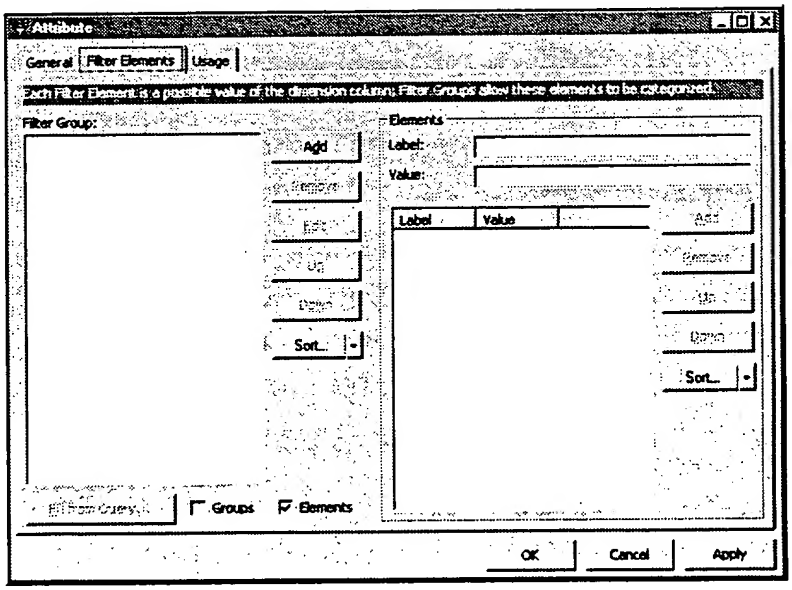
1. Click the **Template** button in the Fill from Query dialog box to display a query template. If you do not want to classify the filter elements into groups, you can use the template query as it is. If you do want to classify filter elements into groups, you can replace the ALL token with an expression or create a union of SELECT statements with different WHERE clauses. Click **OK**.
2. To preview the filter groups produced by your query, select the Filter Elements tab and click **Show Query Results**.

STATIC FILTER GROUPS

When you choose a static filter type, you can use the Filter Elements tab to define filter groups and assign elements to them. Follow these steps to create filter groups for static filter types:

1. Click the **Add** button in the Filter Elements tab to create a new filter group, then enter a label for that group in the Filter Group dialog box. You can also add a glossary entry for the new filter group. (See "Glossary Entries," on page 262.)
2. Continue adding new filter groups. When you are done, you can specify a sort order for the groups as they are to appear on the Filter pop-up Web page by clicking **Sort** and choosing an option from the pop-up menu. You can also move a label to a different place in the list by choosing it and clicking the **Up** or **Down** button.

FIGURE 84: ATTRIBUTE DIALOG BOX: FILTER ELEMENTS TAB



Each Filter Element is a possible value of the dimension column; Filter Groups allow these elements to be categorized.

3. Choose a filter group from the list of filter groups that you just created.
4. You can add filter groups and elements directly by entering attribute values and labels in the **Elements** pane, or you can create a list of elements with an SQL query.

5. To create a list of elements with an SQL query, click the **Groups** and **Elements** check boxes, then click **Fill from Query**.
6. In the Fill From Query dialog box, click **Template** to display an SQL template that you can use to populate the filter group.
7. Edit the SQL template to restrict the column values to only those values you want to include in the current filter group.
8. Click **Test** to verify the query, then click **OK** to assign column values to filter elements in the current filter group. Be sure to append the token `_0$$CURR` to any table names that you add or replace; for example, `Product_0$$CURR`.
9. Update the labels for filter elements. Enter the label for an element in the **Elements** pane, then click the label that you want to update in the Label column. The label updates automatically.

TRANSACTION FILTERS

Transaction filters, which are specific to the List Manager, enable users to filter `group` and `indiv` dimensions by their participation in a fact, such as the purchase of a product. Data related to facts, such as *how much* of an item people bought, is a numeric value, or a measure. To enable end users to apply a filter on measure data in this manner, you need to define transaction filters (via the Web page dialog box) in addition to regular filters. See Chapter 4, "Configuring Web Pages and Topics," for instructions.

ATTRIBUTE FACTORY: FILTERS FOR SURVEYS

For survey purposes, more than one filter on the same attribute may be desired. When extracting schema for survey data, you can use the Attribute Factory dialog box (accessible from a Transaction Filter dialog box) to create transaction filter filters, or two-occurrence filtering. (Since filters are really attributes, you are creating a new attribute, hence the term *Attribute Factory.*)

For example, assume you have an individual base dimension with an `indiv` dimension role that points to a fact table named **Answered**. The **Answered** fact table, in turn, points to the **Answer** dimension table that contains three columns: `Question`, `Answer`, and `Question and Answer Concatenated`. The `Question` column lists each possible question. The `Answer` column lists each answer to each question. The `Question and Answer Concatenated` column shows the distinct combination of each question/answer pair.

TABLE 9: ANSWER DIMENSION TABLE

Question	Answer	Q and A Concatenated
Favorite Color	Red	FavCol:Red
Favorite Color	Blue	FavCol:Blu
Favorite Color	Blue	FavCol:Blu
Favorite Color	Green	FavCol:Gre
Favorite Color	Red	FavCol:Red
Political Affiliation	Republican	PolAff:Rep
Political Affiliation	Democrat	PolAff:Dem
Political Affiliation	Republican	PolAff:Rep
Political Affiliation	Democrat	PolAff:Dem
Political Affiliation	Independent	PolAff:Ind

Assume that an Individual Campaigns Web page has demographic filters for City and Income. The end user wants to further drill down to find out which individuals responded in a certain way to one or more questions, such as those individuals from Omaha with an income in the $50,000 range who are not affiliated with one of the two major political parties and whose favorite color is red (by applying two transaction filter filters).

To enable this type of two-occurrence filtering, design the Web page with a transaction filter named, for example, **Answered**, that consists of the unique question-answer pairs as transaction filter filters. (**Answered** represents a standard transactional filter.)

In this example, the check or list boxes available from the **Answered** transaction filter drop-down list derive from the `Question and Answer Concatenated` column of the **Answer** base dimension table. The query machinery needs to be able to uniquely identify a row; one way to do this is to concatenate question and answer values.

Follow these steps to create multiple transaction filter filters:

1. Open a Transaction Filter dialog box and click the Transaction Filter Filters tab. Click the **Factory** button to open the Attribute Factory dialog box (Figure 85).
2. Select the unique question/answer pair dimension column to be filtered. (`Question and Answer Concatenated` in the example).
3. Select one of these filter types: list box or check box.
4. Click **Template** and fill out the query template.

   The `attribute_name` must be a unique name that is tied to a question. For `<YOUR EXPRESSION>`, enter`['Question'$$CAT` *`question`*`])` where *`question`* is the name of the `Question` column in the **Answer** base dimension table. (The concatenation ensures that the names do not clash with other attributes.) There should be one `attribute_name` per question.

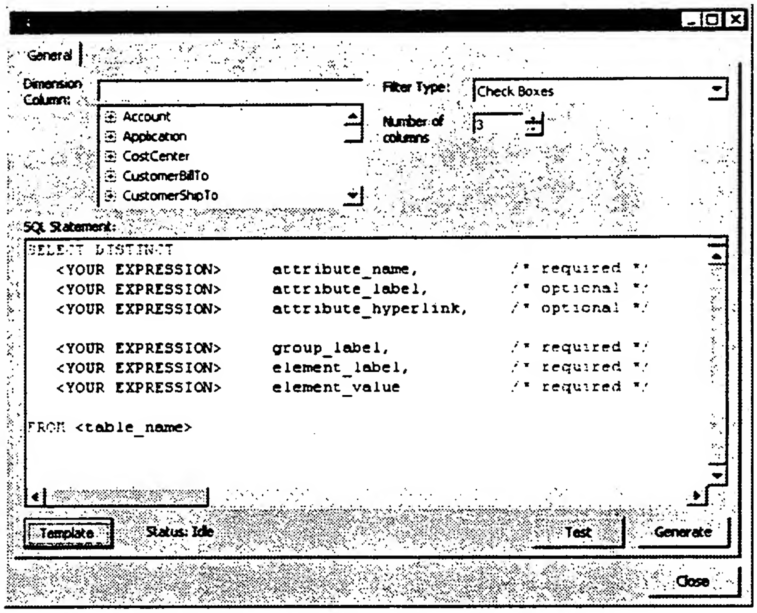
   If appropriate, assign values to `attribute_label`; for example, `attribute_label =` *`question`*, and to `attribute_hyperlink`.

   The filter `group_label` is a further grouping mechanism. It is only for subcategories, for example, if one wanted to group Political Affiliation by categories such as frequent or infrequent voters. It can also be a literal, such as `All`. (A filter group label applies only to a check-box filter type.)

   For the filter `element_label`, enter the name of the `Answer` column in the **Answer** dimension table. These are the labels of the individual columns in the **Answer** drop-down list on the Web page.

For the filter `element_value`, enter the name of the unique question/answer pair column.

FIGURE 85: ATTRIBUTE FACTORY DIALOG BOX



```
SELECT DISTINCT
   <YOUR EXPRESSION>     attribute_name,          /* required */
   <YOUR EXPRESSION>     attribute_label,         /* optional */
   <YOUR EXPRESSION>     attribute_hyperlink,     /* optional */

   <YOUR EXPRESSION>     group_label,             /* required */
   <YOUR EXPRESSION>     element_label,           /* required */
   <YOUR EXPRESSION>     element_value            /* required */

FROM <table_name>
```

SECURITY

The E.piphany system provides two areas of security:

- *Authentication*, or a user's ability to log on to the system. Authentication is determined by the Windows NT operating system.
- *Access rights*, or the permissions a user has after logging on. Access rights include the user's ability—
  - to open a Web page associated with a topic/navigation node.
  - to perform a specific navigation step.
  - to save queries for those Web pages. Saved queries (called *reports*) are lists of the options that a user selected on a Web page (to generate a report).

- to access data based on the values of dimensional attributes; that is, to restrict access rights to the Web page to certain dimensional attribute values. You can use this kind of security to allow some users to see data for one region but not another.

  The E.piphany system can also use Windows NT groups to administer access rights.

You can use the **Security/Storage** folder in the EpiCenter Manager directory tree to add groups and users to the system. The two work in tandem: groups have users, and users belong to groups. When setting up a new E.piphany system, first set up groups and then add members to them after defining users. (Windows NT groups members can simply be imported.)

The **Security/Storage** folder also contains the Report Gallery, which enables the administrator to organize all saved reports in folders for groups and users. See "Report Gallery," on page 287 for more information.

A group can be marked administrative. Users that are members in this group are administrative users that have these special privileges:

- access to all folders and reports
- all navigation paths are enabled
- all column restrictions are disabled
- monitor functionality is exposed to administrative users only

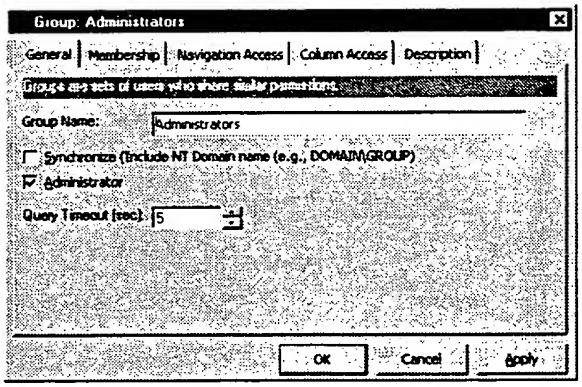
SETTING UP GROUPS

Members of a group share similar access rights, or permissions. In general, you should attach these to groups instead of users to simplify maintenance. Access rights set for a user, however, take precedence over access rights set for groups to which the user belongs.

To define a group, right-click the **Group** folder and select **New Group** from the pop-up menu. (This menu also has commands for exporting all groups, importing NT Groups, and refreshing the group folder list.) Complete the information requested in the Group dialog box tabs (General, Membership, Navigation Access, Column Access, and Description) as described in this section. (Use the Description tab to document the group for your own reference.)

1. In the General tab (Figure 86), enter the group's name.

FIGURE 86: GROUP DIALOG BOX: GENERAL TAB



Groups are sets of users who share similar permissions.

Group Name: Administrators

□ Synchronize (Include NT Domain name (e.g., DOMAIN\GROUP)

☑ Administrator

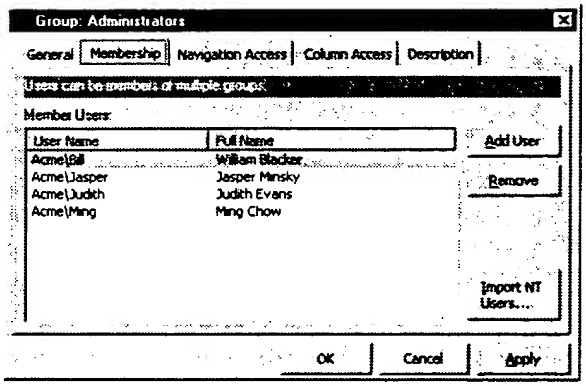
Query Timeout (sec): 5

2. Select the **Synchronize** option to add new users to the group in an "autopilot" fashion via Windows NT synchronized groups. You need to precede Windows NT synchronized group names with their Windows NT domain name prefix (and matching name).

   Each time a user logs in, the Windows NT security API is accessed for a list of group names to which that user belongs. If the user is a member of an NT group that also exists in the Epiphany world and is marked synchronized, and there is no membership record for that user and that group in EpiCenter Manager, then a new record is created. Conversely, if the user is no longer a member of an NT group, but there is such a membership record in EpiCenter Manager, and the group is marked synchronized, then the latter record is removed.

3. Select **Administrator** if members of this group have administrative rights. An Administrator can see all Web pages and reports in the system, but cannot modify special folders, such as the **Public** folder. (See "Report Gallery," on page 287 for a discussion of special folders.)
4. Enter the seconds for the maximum time that a query should run for members of this group. As an administrator, you may set a time-out limit to ensure that someone does not monopolize database engine resources.
5. You can use the Membership tab (Figure 87) to add already created users to the group (as described in "Setting Up Users," on page 279). Users may be members of multiple groups. Click **Add User.**

FIGURE 87: GROUP DIALOG BOX: MEMBERSHIP TAB



Group: Administrators

Users can be members of multiple groups.

Member Users:

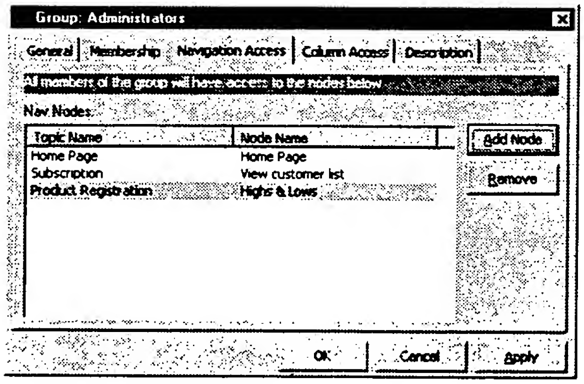
User Name	Full Name
Acme\Bill	William Blacker
Acme\Jasper	Jasper Minsky
Acme\Judith	Judith Evans
Acme\Ming	Ming Chow

6. Select the user's name from the Choose User dialog box and click **OK**, or click **New**, which displays the User dialog box.
7. To import NT users into the group, click **Import NT Users** and enter the domain in the dialog box. You have the option of adding the new users to all E.piphany groups in which they are members of corresponding NT groups. This actions adds users to existing groups; it does not add new E.piphany groups.

8. You can use the Navigation Access tab to assign all members of the group access to topics and navigation nodes within topics. Before E.piphany users can open a Web page, they must be granted access to its associated topic/node. Users have the ability to access all nodes granted access to their groups by the administrator. Click **Add Nodes** and select the topic/node from the list.

Note: *Before you can configure solutions or create new topics, permissions must be set for all nodes. Otherwise, these nodes do not appear when you start the AppServer. You can select all nodes when adding them to a user or group. Alternatively, you can open the topic, select the Navigation Nodes tab and multi-select all of the nodes in the list. Click **Add Group**, and choose the group to assign all members of the group permission to see all of the nodes in the topic.*

FIGURE 88: GROUP DIALOG BOX: NAVIGATION ACCESS TAB



Group: Administrators

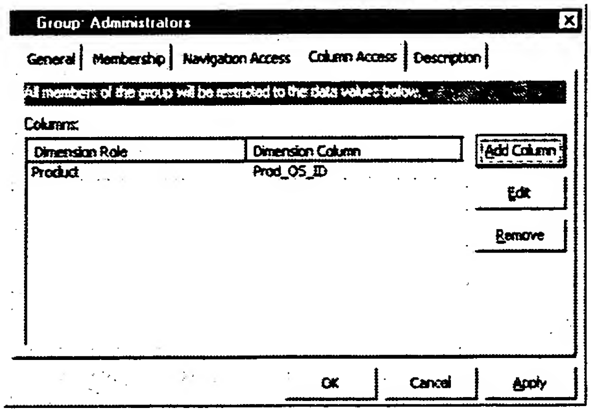
All members of the group will have access to the nodes below

Nav Nodes:

Topic Name	Node Name
Home Page	Home Page
Subscription	View customer list
Product Registration	Highs & Lows

You can use the Column Access tab (Figure 89, on page 278), which is the same for users and groups, to restrict access rights for the Web page to certain attributes (dimension columns).

Figure 89: Group Dialog Box: Column Access Tab



Group: Administrators

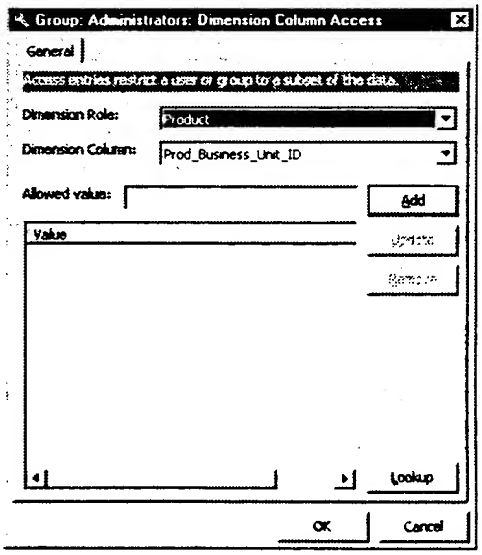
All members of the group will be restricted to the data values below.

Columns:

Dimension Role	Dimension Column
Product	Prod_OS_ID

1. Click the **Add Column** button, which displays the Dimension Column Access dialog box (Figure 90).

Figure 90: Dimension Column Access Dialog Box



Group: Administrators: Dimension Column Access

Access entries restrict a user or group to a subset of the data.

Dimension Role: Product

Dimension Column: Prod_Business_Unit_ID

Allowed value:

2. Select the Dimension Role for this dimension column from the drop-down list.

3. To display all of the values in the dimension column, click **Lookup**, which directly accesses the EpiMart data. Select a value or values from the listing and click **OK** to add it to the value list.

   The values entered in this list should correspond to actual database values in the base dimension table to which that dimension column corresponds. For example, selecting `Date.fy_name` as the field with the values 1997 and 1998 causes all reports to be filtered with these values.

4. If you know the exact value of the dimension column whose values are accessible to the group, type it in the **Allowed values** text box, and click **Add** to place it in the **Value** listing below. Repeat this step to add additional values.

5. Click **OK**.

After a group has been defined, it becomes a sub-folder of the **Group** folder. You can right-click a group folder and use the pop-up menu to set up a new group, or to edit, delete, or duplicate the group. A duplicate group is identical except for its group name.

SETTING UP USERS

All authorized users of the E.piphany system appear as icons in the **Users** sub-folder of the **Security/Storage** folder. By default, new users have these permissions:

- no access to any topics/nodes.
- no group membership.
- inability to save any reports.
- access to all data in all dimensions—there are no dimensional attribute (column) restrictions.

For column restrictions only, the access rights set for a user have precedence over access rights set for groups to which the user belongs.

To set up a new user, right-click the **Security/Storage** folder and select **New User**. The User dialog box (Figure 91) that displays has tabs for General, Membership, Navigational Access, and Column Access.

Configure the General tab as follows:

1. Enter the person's username and first and last names.

   The first and last names are used for display purposes only. If they are missing, then the username is displayed.

   For Windows NT authentication, enter users with their Windows NT domain prefix to distinguish among users with the same name but who are in different domains.

2. Select the Report Gallery type (view mode) for this user from the drop-down list.

   There are three Report Gallery modes:

   - **FoldersAndFiles**, which shows both folders and files. It allows you to switch to **Files** mode.
   - **Files**, which shows all files that the user has access to. It allows you to switch to **FoldersAndFiles** mode.
   - **FoldersAndFilesOnly**, which shows both folders and files. You cannot switch out of this mode.

3. Check whether the user should go directly to the results of a report for gallery, mailed, and favorite reports. If executed is not checked for one of these options, the user is sent to the query parameters on the Web page instead of the report itself (and must generate the results).

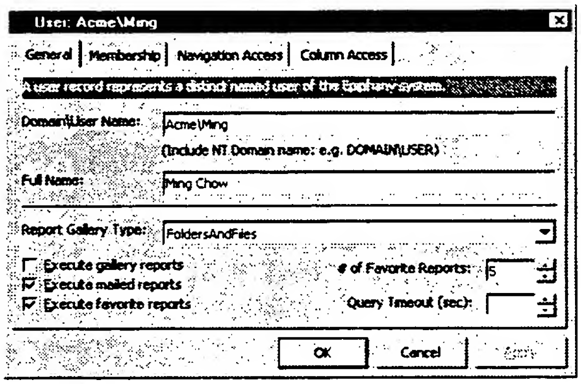
Note: *A user can e-mail another user a report by typing the URL in the text, which the recipient can then click to open the report in a browser window.*

4. Indicate the maximum number of favorite reports that this user may have.

   The user can choose how many favorite reports appear below each topic on a home page. Although there is no maximum number, five reports per topics is recommended.

5. Enter the seconds for the maximum time that a query should run for this user. As an administrator, you may set a time-out limit to ensure that a user does not monopolize database engine resources.

FIGURE 91: USER DIALOG BOX: GENERAL TAB



A user record represents a distinct named user of the Epiphany system.

Domain\User Name: Acme\Ming

(Include NT Domain name: e.g. DOMAIN\USER)

Full Name: Ming Chow

Report Gallery Type: FoldersAndFiles

□ Execute gallery reports

☑ Execute mailed reports

☑ Execute favorite reports

# of Favorite Reports: 5

Query Timeout (sec):

After groups have been created as described in “Setting Up Groups,” on page 274, you can use the Membership tab to assign group memberships to the user. Click **Add Group** and select one or more groups from the Choose Groups dialog box. (Hold down the Shift key to select more than one.)

A user can be a member of multiple groups, but for security reasons needs to be a member of a a single group (called the primary group) when running queries. The restrictions set for the primary group apply to all of the user’s queries. After assigning the user group memberships, select the main group in the list and then select the **Primary** option in the tab.

Note: *Adding users to multiple groups makes it easier for users to share saved reports. A user may save reports into any group folder for which he or she is a member.*

If there is no group that contains all of the user’s privileges, or if the user has privileges that apply only to that user and no other, assign privileges to the user. Individual privileges that you set always override group privileges, even if the user is a member of a primary group.

Before E.piphany users can open any Web page, they must be granted access to its topic and node. Users have the ability to see and use all nodes to which they have access, and to all nodes granted access to their groups. In the Navigational Access tab, click **Add Nodes** and select the Nodes in the list that the user may access. Hold down the **Shift** key to make multiple selections.

You can use the Column Access tab to restrict access rights for the Web page to certain attributes (dimension columns). Follow these steps:

1. Click **Add Column**, which opens the Dimension Column Access dialog box (Figure 90, on page 278.
2. Select the Dimension Role for this dimension column from the drop-down list.
3. To display all of the values in the dimension column, click **Lookup**, which directly accesses the EpiMart data. Select a value or values from the listing and click **OK** to add it to the value list.

   The values entered in this list should correspond to actual database values in the base dimension table to which that dimension column corresponds. For example, selecting `Date.fy_name` as the field with the values 1997 and 1998 causes all reports to be filtered with these values.
4. If you know the exact value of the dimension column whose values are accessible to the group, type it in the **Allowed values** text box, and click **Add** to place it in the **Value** listing below. Repeat this step to add additional values.

STORAGE

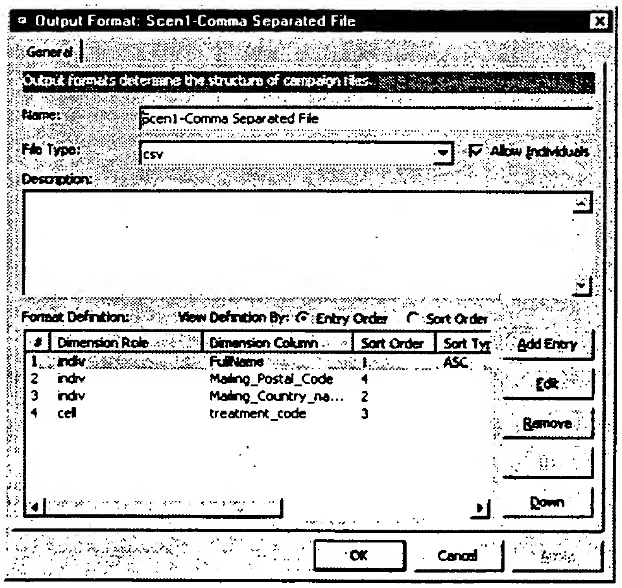
In addition to the security settings folders, the **Security/Storage** folder also contains folders for campaign output and report configuration.

OUTPUT FORMATS

An output format specifies the structure of the files generated in a campaign. To define a new output format:

1. Right-click the **Output Formats** folder and select **New Output Format**.
2. In the **Output Format** dialog box (Figure 92), enter a name and a description for the output format.

FIGURE 92: THE OUTPUT FORMAT DIALOG BOX



Output Format: Scen1-Comma Separated File

Output formats determine the structure of campaign files.

Name: Scen1-Comma Separated File

File Type: csv ☑ Allow Individuals

Description:

Format Definition: View Definition By: ● Entry Order ○ Sort Order

#	Dimension Role	Dimension Column	Sort Order	Sort Typ
1	indiv	FullName	1	ASC
2	indiv	Mailing_Postal_Code	4	
3	indiv	Mailing_Country_na...	2	
4	cell	treatment_code	3	

3. Choose an output file type from the drop-down list. The following file types are available:
   - **`csv`** — Comma-separated values (text format)
   - **`html`** — HTML
   - **`tsv`** — Tab-separated values (text format)

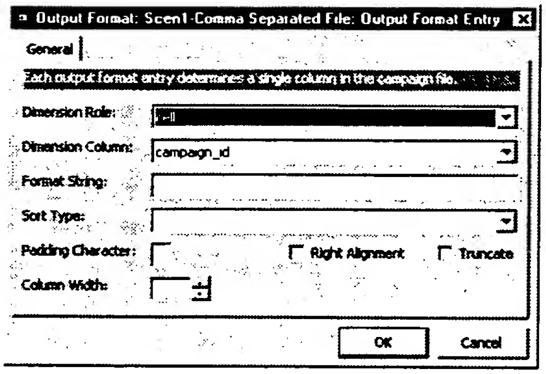
- **`txt`** — Text (text format)
- **`xls`** — Excel spreadsheet

4. Check **Allow Individuals** if this format should allow attributes for both individuals and groups. (If this is not checked, then individual attributes cannot be specified in the format.)
5. Add the entries to be included in the output.

To add an entry:

1. Click **Add Entry**, which displays the **Output Format Entry** dialog box (Figure 93).

FIGURE 93: THE OUTPUT FORMAT ENTRY DIALOG BOX



Output Format: Scen1-Comma Separated File: Output Format Entry

General

Each output format entry determines a single column in the campaign file.

Dimension Role: Cell

Dimension Column: campaign_id

Format String:

Sort Type:

Padding Character:

☐ Right Alignment

☐ Truncate

Column Width:

2. If the entry is a dimension column value, select the dimension and column from the drop-down list. If this does not apply, you can use the **Format String** text box.
3. In the **Format String** text box, if the entry is text format (`txt`, `csv` or `tsv`), enter a standard `java.text.MessageFormat` formatting string. See your Java Development Kit (JDK 1.1) documentation for instructions. The URL is:

```
http:// java.sun.com/products/jdk/1.1/docs/api/
java.text.MessageFormat.html
```

For example, the format string `Year {0}` translates to the value of `Year 1997` when the value of the dimension column in the database is 1997. The format `{0,number,percent}` translates to the value of `35.5%` when the value of the dimension column is 0.355.

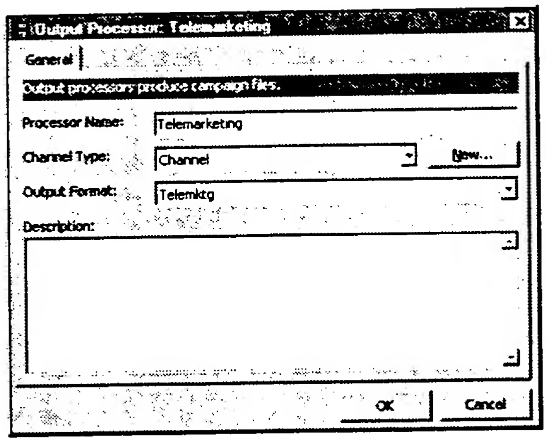
4. For `html` or `xls` file types, enter the name of the column header without quotes. The backslash (\) character is special, as long as it is followed by either `n`, `r`, or, `t`; as in `\%` (where `%=n,r,t`). If the backslash is not followed by one of these three characters, it is ignored. A double backslash (\\) is interpreted as a single (\).
5. Select one of the sort types from the drop-down list.

   If nothing is set, then the records in the file are not sorted. If **ASC** or **DESC** is selected, then records are sorted in ascending or descending alphabetic order, respectively. **EXCLUDE_ASC** or **EXCLUDE_DESC** are used for sorting records internally, not for sorting output.
6. For text formats, enter any padding characters to be added to the entry record. These characters fill out the column to the specified column-width.
7. Check **Right Alignment** if you want the entry to be padded to the left of the record. If unchecked, padding occurs to the right.
8. Check **Truncate** if you want the field to be truncated if it is greater than the column width.
9. For text formats only, select column-width options. A column width of 0 indicates an arbitrary column width.
10. Click **OK** to add the entry and return to the **Output Format** dialog box.
11. If multiple fields are to be sorted, select **Sort Order** in **View Definitions By.** Move entries to the appropriate location using the **Up** and **Down** buttons.
12. Select **Entry Order** in **View Definitions By** to view the order in which the entries appear in the output.

OUTPUT PROCESSORS

An output processor specifies an output option that is available to the end user. Output processors make use of output formats, so you need to define an output format in order to define an output processor.

FIGURE 94: THE OUTPUT PROCESSOR DIALOG BOX



Output Processor: Telemarketing

General

Output processors produce campaign files.

Processor Name: Telemarketing

Channel Type: Channel

Output Format: Telemktg

Description:

To define an Output Processor:

1. Right-click the **Output Processors** icon and choose **New Output Processor.**
2. In the Output Processor **dialog box** (Figure 94), enter a name and description for the output processor.
3. Choose a channel type from the drop-down list. Use the **New** button to define a new channel.

   A channel is a category of processors grouped by the type of campaigns that they execute; for example, Direct Mail, Telemarketing, Web Site, and so forth.
4. Choose an output format from the drop-down list.
5. Click **OK** to define the output processor.

REPORT GALLERY

Note: *The Report Gallery in EpiCenter Manager is for administrative use only. Many of the features of this Report Gallery can also be accomplished by end users using the Report Gallery in the Web interface.*

You can use EpiCenter Manager to browse the saved reports for a Web page, as well as view who has access to those reports. All of the saved reports for the E.piphany system are organized in folders and saved reports in the Report Gallery. When a user saves a report, it is saved in whatever folder the user is currently in. This can be any folder to which the user has write access.

The Report Gallery has a top-level folder that contains the **Public** folder, **All Users** folder, and **All Groups** folders. Everyone has access to **Public** folder by default. Any user can save reports there. There is a folder for each user under the **All Users** folder. There is also a folder for each group under the **All Groups** folder. Every user folder, group folder, and public folder has three sub-folders: **Defaults**, **Favorite Reports**, and **Favorite Charts**.

A default report is the set of Web page settings that a user sees when he or she first opens that Web page. In general, the most specific default report is used. Thus if a user-level default report exists, that report is opened in the user's Web browser. If no user default report exists but a group default report exists for the Web page, the group default report is displayed. The default report in the **Public** folder should always be available if there is no specific user or group default report.Whenever a new Web page is created, the administrator should save one report in the Default sub-folder in the Report Gallery's **Public** folder and make it the default.

When a report is saved in a **Favorite Reports** folder, it shows up on the home page under the topic for that report.

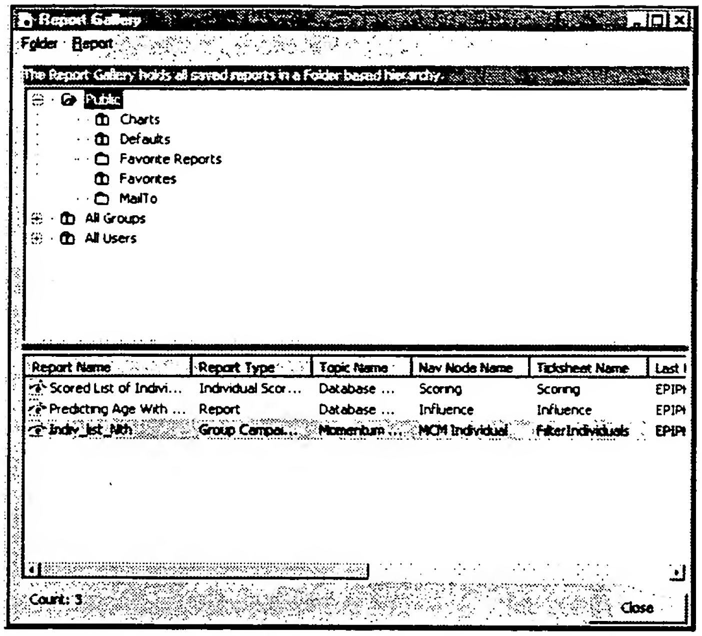
If you enlarge a chart on the results page, you can save it in the **Favorite Charts** folder, which shows the chart on the home page. (The chart is also scheduled for updating in the Charts queue.)

Using EpiCenter Manager, the administrator can control the access to the Report Gallery's folders. For example, modifying the access to the public **Defaults** folder determines who can save public defaults (by default everybody can save public defaults). Turning off all permissions for the **Public** folder disables Public folder completely. Turning off permissions for the **All Groups** folder disables the **All Groups** folder completely. Do not remove the folders. Modify permissions on them if you want to control the access to them.

Note: *When you change the names of attributes, transaction filters, output processors, topics, navigation nodes, options, measures, measure sets, and Web pages, you are requested to confirm the name change. These are unique name fields in metadata objects that saved reports reference internally. If these names are changed, then parts of the saved report may become invalid. For example, if an attribute has been renamed, and you open a saved report that uses that attribute and attempt to execute it, you receive an error message because the attribute is invalid. If, however, you open the saved report without executing it, the Web page has the default attribute. You can simply re-save the report with the new attribute.*

To display the Report Gallery, right-click the **Security/Storage** folder and select **Report Gallery** from the pop-up menu. The Report Gallery (see Figure 95) is displayed. The top pane shows the folders organized in a tree hierarchy, and the lower pane lists the reports for a selected folder by report name, report type, topic name, navigation node name, Web page name, date last modified, and the person who modified it.

FIGURE 95: REPORT GALLERY IN EPICENTER MANAGER



The Report Gallery holds all saved reports in a Folder based hierarchy.

- Public
  - Charts
  - Defaults
  - Favorite Reports
  - Favorites
  - MailTo
- All Groups
- All Users

Report Name	Report Type	Topic Name	Nav Node Name	Ticksheet Name	Last I
Scored List of Indivi...	Individual Scor...	Database ...	Scoring	Scoring	EPIPH
Predicting Age With ...	Report	Database ...	Influence	Influence	EPIPH
Indiv_list_Nth	Group Campai...	Momentum ...	MCM Individual	FilterIndividuals	EPIPH

Count: 3

The Report Gallery's main menu has menus for **Folder** and **Report**, which are described below.

FOLDER MENU

Note: *Right-clicking in the top pane of the Report Gallery displays a pop-up menu with the Folder commands.*

You can use the **Folder** menu to:

- create a folder for saved reports. Choose **New** from the **Folder** menu, select the folder or sub-folder, and enter the new folder name in the dialog box.
- delete a selected folder.
- display the folder's properties. (You can also double-click the leaf folders to open the Folder dialog box.)

  The Folder dialog box has two tabs: General and Permissions. The General tab shows the folder's name, path, date last modified, and any description.

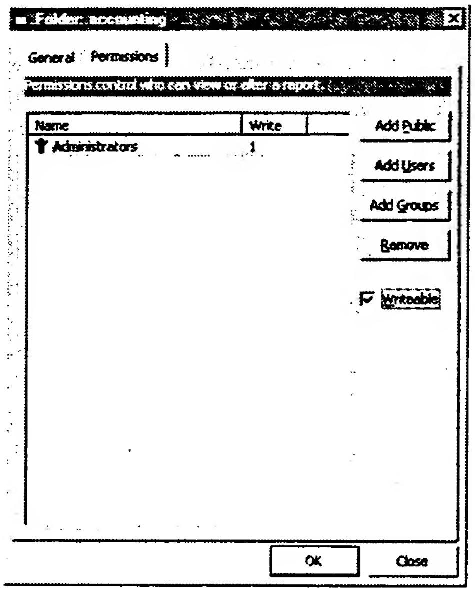
The Permissions tab allows you to set controls for who can view and alter the folder. See “Setting Permissions,” on page 292 for more information. *This is the only way to modify public defaults.*

- find a report in one of the folders.

  Enter the file name in the Find Report dialog box. You can refine your selection to report types and date modification ranges. Pressing the **F3** key locates the next report that meets this criteria.

- move folders by dragging them.

FIGURE 96: FOLDER DIALOG BOX: PERMISSIONS TAB



Folder: accounting

Permissions control who can view or alter a report.

Name	Write
Administrators	1

☑ Writeable

REPORT MENU

Note: *Selecting a report file in the lower pane displays a pop-up menu with the Report commands.*

You can use the **Report** menu to copy and move reports and folders. (This functionality is not available through the Web interface Report Gallery.)

To copy a report:

Select it, and choose **Copy** from the **Report** menu. Select the folder in which you would like to place the copy of the report, and select **Paste**.

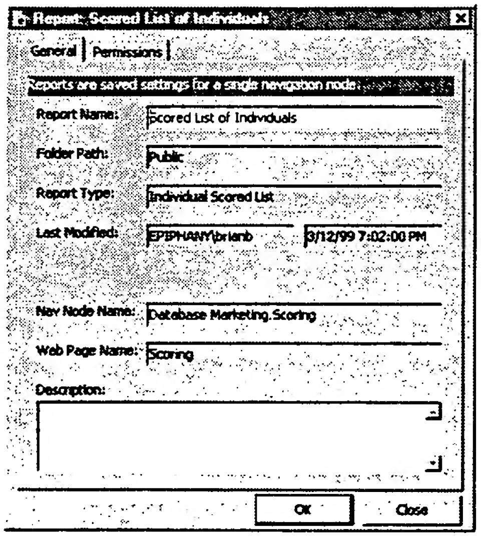
You can also cut and paste a report, or explicitly delete it.

Double-clicking a report opens the Report dialog box (Figure 97).

- The General tab shows the report's name, folder path, report type, Web page name, date last modified, and any description.
- The Permissions tab allows you to set controls for who can view and alter the report. See Setting Permissions for more information.

  When a user saves a report without explicitly setting permission, default permissions are assigned. See "User Default Permissions" on page 292.

FIGURE 97: REPORT DIALOG BOX



Report: Scored List of Individuals

Reports are saved settings for a single navigation node.

Report Name: Scored List of Individuals

Folder Path: Public

Report Type: Individual Scored List

Last Modified: EPIPHANY\brianb 3/12/99 7:02:00 PM

Nav Node Name: Database Marketing.Scoring

Web Page Name: Scoring

Description:

SETTING PERMISSIONS

You can use the Permissions tab of the Folder and Report dialog boxes (see Figure 96) to assign users and groups permission to view a folder or report.

If you select **Writable** in the Report Permissions tab before clicking the **Add** button, then the users or groups you add may alter the report. Selecting **Writable** in the Folder's Permissions tab gives write access to any report in the folder for the user or group (although a user cannot create a new report in a folder unless he or she has write access to that folder).

- To allow all users the ability to view this folder or report, click **Add Public.**
- To give users access to folders/reports, click **Add Users** and select users from the Choose dialog box.
- To give groups access to folders/reports, click **Add Groups** and select groups from the dialog box.

USER DEFAULT PERMISSIONS

When a user saves a report without explicitly setting permission, default permissions are assigned. Default permissions are calculated as follows:

- If the user creates a report in a regular folder, the folder's permissions are used.
- If the user creates a report in a user folder, default permissions on the report are Read/Write for the user only.
- If the user creates a report in a group folder, default permissions on the report are Read/Write for the group only.

By default, others have read permission on a user's folder. A user cannot set up his or her user folder (private folder) in such a way that other users can create, delete, or update properties on folders or reports in that folder. It is possible, however, for users to overwrite reports that they have access to in the user folder.

To create, delete, or update a report or folder, a user must have write permission to the folder in which the operation is about to be performed (the parent folder). Overwriting a report requires read-permission-only for the parent folder (not write permission).

GENERATING SCHEMA

After defining your metadata via EpiCenter Manager, you need to convert these definitions into actual tables (the EpiMart). You can use the Generate Schema dialog box to build the EpiMart tables, as well as to populate the physical date dimension table. The Date dimension table is a special base dimension table supplied by E.piphany for storing all attributes related to time. All fact tables in an EpiCenter receive a foreign key (called `date_key`) to this table.

To generate the schema for your EpiMart:

1. Choose **Generate Schema** from the **EpiCenter** menu. The Generating EpiMart Schema dialog box has two tabs: Schema and Results.
2. In the upper pane of the Schema tab, select **Trial Run** to see what the results are without making permanent changes to the files.
3. By default, only tables that have changed since the last time you generated schema are rebuilt. Select **Force Rebuild of all Tables** only if you want to rebuild *all* of your tables.
4. To generate schema for the Campaign Manager-related tables, check Include Backfeed Tables.
5. Check **Build Sampling Table**. The sampling table is a master table (a list of random numbers) that E.piphany programs use to produce samples of dimension tables. Select the number of rows to be populated in the sample dimension tables.

Set this value to *at least* the larger value of–

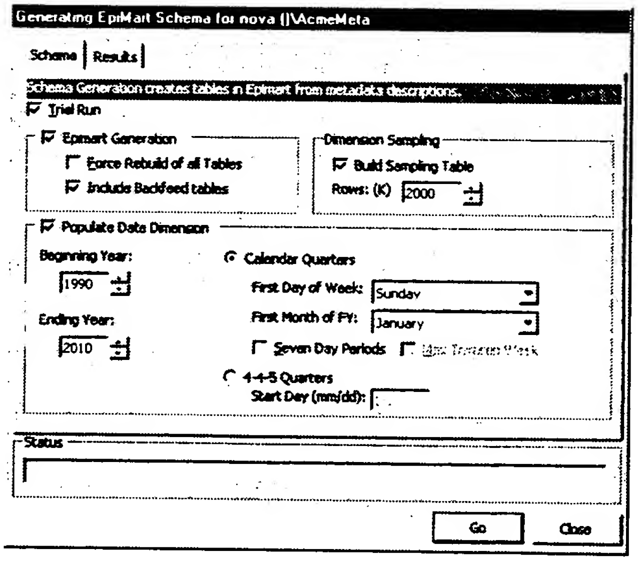
- the number of rows in the individual dimension, or
- the number of rows in the group dimension.

This value should be larger than the number of rows in the largest dimension table.

6. The date dimension table must be populated the first time you generate schema. Follow the instructions for populating the data dimension below.
7. Click **Go** (and watch the progress bar).
8. After the trial run is finished, click the Results tab to view which tables were updated.
9. If the results are acceptable, de-select **Trial Run** in the Schema tab, and click **Go**.

After the schema has been generated, the system records the current state of the metadata so that the next time the schema is updated, the current definitions can be compared with the new ones, and the appropriate tables can be created or altered as necessary.

FIGURE 98: GENERATING EPIMART SCHEMA DIALOG BOX



Schema Generation creates tables in Epimart from metadata descriptions.

☑ Trial Run

☑ Epimart Generation
- ☐ Force Rebuild of all Tables
- ☑ Include Backfeed tables

Dimension Sampling
- ☑ Build Sampling Table
- Rows: (K) 2000

☑ Populate Data Dimension
- Beginning Year: 1990
- Ending Year: 2010
- ● Calendar Quarters
  - First Day of Week: Sunday
  - First Month of FY: January
  - ☐ Seven Day Periods ☐
- ○ 4-4-5 Quarters
  - Start Day (mm/dd):

Status

To populate the date dimension table:

1. Select **Populate Date Dimension** in the Schema tab of the Generating EpiMart Schema dialog box. (You can also choose **Populate Date Dimension** from the **EpiCenter** menu to populate the dates only; see "Populating the Date Dimension," on page 151.)
2. Enter the values for the beginning and ending years of the EpiMart. The date range of the EpiMart should be at least as large as any dates that are found in the data you are extracting. These values are defined in the Configuration dialog box (choose **Configuration** from the **EpiCenter** menu to open it). For users to obtain the best results when forecasting trends, you need to build the date dimension as far into the future as you plan to forecast. (Currently, the maximum prediction is three years past the last date that has recorded data.) If the date dimension is not built out far enough, the user receives columns with names such as:

   ```
   Dec 1999, Second Next, Third Next
   ```

   instead of:

   ```
   Dec 1999, Dec 2000, Dec 2001
   ```

3. Choose the appropriate quarter system: Calendar Quarters (three month divisions of the year) or Quarters 4-4-5. *4-4-5 Quarters* represents the 13-week-per-quarter calendar in which the months in the quarter are defined as consisting of 4 weeks, 4 weeks, and 5 weeks.
4. *For Calendar Quarters*, enter the first day of the week and the first month of the fiscal year. Select **Seven Day Periods** to ensure that all fields that count up in number (such as `week_number_fq`) begin with seven day period; for example:

   ```
   111111122222223333333...13 or 14
   ```

   whereas all fields that count down in number, such as `week_number_til_end_gq`) end with seven day periods; for example:

   ```
   14 or 13...333333322222221111111
   ```

   The alternative is that weeks may overlap quarter boundaries incompletely.

5. For **Calendar Quarters**, select **Max Thirteen Week** to prevent a "14th" week or "53rd" year; the 13th or 52nd weeks are simply extended as needed.
6. For **4-4-5 Quarters**, enter the start date of the first quarter.

Appendix C, "Date Dimension Fields," describes the date dimension fields used by the E.piphany system.

EXPORTING/IMPORTING METADATA

E.piphany provides a metadata export/import utility for moving metadata between EpiCenters and for backing up the definition of an EpiCenter. To use this tool properly, you need to understand the metadata concepts (see Metadata Overview in Appendix G, "Export/Import of Metadata").

Note: *Within the EpiCenter tree in a single EpiCenter Manager window, you can drag and drop items or folders from one EpiCenter to another.*

The export operation produces a Microsoft Access database that contains a representation of the metadata that was chosen for export. Upon import, this representation is converted back into valid E.piphany metadata in the target EpiMeta. Using EpiCenter Manager, you have granular control over which metadata objects get exported during each operation. For example, you can use the same objects and Web pages in several EpiCenters. Also, during import, you control whether or not to overwrite existing metadata that conflicts with what is in the import file.

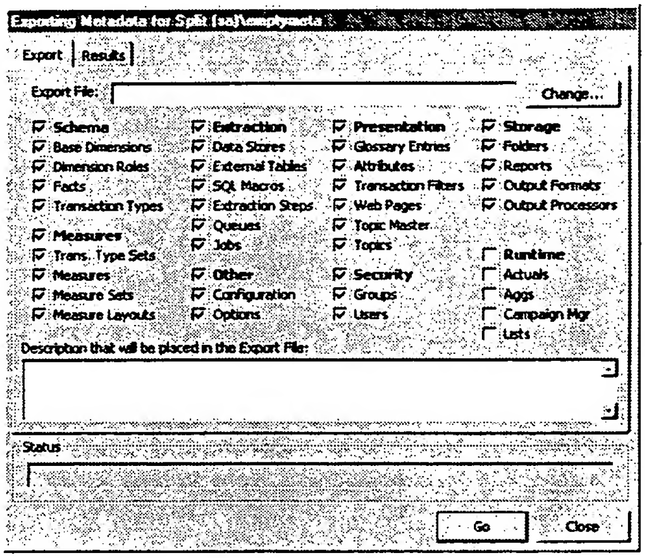
Reasons for exporting files other than publishing them for import are to save files for document control (source safe) and to send files as an e-mail attachment; for example, you might e-mail an exported file to Technical Support for analysis.

You can export individual metadata objects by right-clicking folders in the EpiCenter Manager directory tree and selecting the **Export** command from the pop-up menu. This option is available only for exportable metadata objects.

To display a dialog box in which you can export all or any subset of metadata files:

1. Right-click the EpiCenter icon and select **Export\Export All** from the pop-up menu, which displays the Exporting Metadata dialog box (Figure 99, on page 297).

FIGURE 99: EXPORTING METADATA DIALOG BOX



Exporting Metadata for Split (sa)\emptymeta

Export | Results

Export File:

☑ Schema
☑ Base Dimensions
☑ Dimension Roles
☑ Facts
☑ Transaction Types

☑ Measures
☑ Trans. Type Sets
☑ Measures
☑ Measure Sets
☑ Measure Layouts

☑ Extraction
☑ Data Stores
☑ External Tables
☑ SQL Macros
☑ Extraction Steps
☑ Queues
☑ Jobs

☑ Other
☑ Configuration
☑ Options

☑ Presentation
☑ Glossary Entries
☑ Attributes
☑ Transaction Filters
☑ Web Pages
☑ Topic Master
☑ Topics

☑ Security
☑ Groups
☑ Users

☑ Storage
☑ Folders
☑ Reports
☑ Output Formats
☑ Output Processors

☐ Runtime
☐ Actuals
☐ Aggs
☐ Campaign Mgr
☐ Lists

Description that will be placed in the Export File:

Status

2. In the Export tab of the Exporting Metadata dialog box select a heading, such as **Schema** to select all of the Schema topics automatically, or select individual subtopics.

3. Enter any description to be placed in the export file. This description can be read in the Importing Metadata dialog box when the user selects this file for import.

4. Click **Go**.

To import individual metadata files in EpiCenter Manager, right-click a folder and select **Import** from the pop-up menu. This option is available only for metadata files that can be imported.

You can produce a new EpiCenter by importing all of the metadata files from an existing EpiCenter. To import all of the metadata:

1. Right-click the EpiCenter icon and select **Import Metadata** from the pop-up menu.
2. The default export file is **export.mdb** in your **ConfigFiles** directory. Click **Change** to select another file.
3. Select **Always Replace Existing Data** if you want the new data to replace existing entries of the same name.
4. Select **Continue after Errors** so that should an error occur, you can receive a partial report. To prevent the importation of excessive errors, the **Maximum [number of errors] Not Imported** is set to 100 by default.
5. Click **Go**.

Note the following:

- Re-importing objects does not delete references to those objects. For example, a measure definition can be re-imported without deletion of the measure mappings in Web pages that point to that measure.
- To speed up import and export, the entire Microsoft Access database is either read from or written to at once.
- All import operations are performed inside of a single database transaction. Any error that occurs during import can be rolled back completely.

See Appendix G, "Export/Import of Metadata" for more information.

TOGGLING DATAMART TABLES

To switch to the A set of tables from the B set, or vice versa, choose **EpiCenter\Toggle Datamart**, or click the **A or B** tables toolbar button. Queries run against the set of tables indicated by the enabled button, and extractions populate the set indicated by the disabled button.

The current datamart is shown at the bottom of the EpiCenter Manager window. See “Mirroring and History: A, B, X and Y Tables,” on page 96 for more information.

RUNNING THE SCRUTINY DEBUGGING TOOL

Scrutiny is an interactive debugging tool available within EpiCenter Manager. (It is also the command-line executable **scrutiny.exe**.) Scrutiny performs a superset of checks that the E.piphany Application Server performs at startup time. Scrutiny provides output that you can examine instead of the AppServer (**SRV**) log file. If Scrutiny finds a problem, it attempts to fix it.

Scrutiny ensures that your EpiCenter (both EpiMeta and EpiMart) is in a consistent and functioning state. It does this by running an extensive set of queries against your EpiMeta and EpiMart to ensure various rules are followed and multiple descriptions of items are consistent. For example, Scrutiny can examine the EpiMart tables for missing indexes, missing rows, bad referential integrity, and so forth. It can also catch List Manager constraints such as not allowing group transaction filters for individuals. It also checks consistency constraints, such as the presence of **UNKNOWN** rows in EpiMart tables and well-formed references.

You can run Scrutiny periodically as a validation tool, or whenever you encounter any problems with metadata or EpiMart data, or if the Application Server fails to start. Scrutiny diagnoses and fixes many common (and less common) issues.

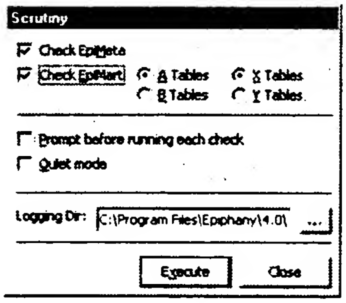
Choose **Run Scrutiny** from the **EpiCenter** menu. In the Scrutiny dialog box (Figure 100), select whether you want to perform EpiMeta or EpiMart checks, or both. Select the EpiMart tables you would like to check: A/B or X/Y. (By default, the value of the current datamart is selected.) You can select to have the program prompt you before issuing each type of check. In quiet mode, only errors and dots that mark Scrutiny's progress are displayed.

In the **Logging Dir** text box, the initial default logging directory is the location of EpiCenter Manager directory. You can enter a new directory in the text box, or click the button to browse for one. The E.piphany system stores the last directory you have logged to in the Registry, and this becomes the new default.

Because many Web pages are incomplete for end use until options are set for their default report, Scrutiny checks for navigation nodes that do not have default reports. When Scrutiny runs, it displays warnings regarding any missing default reports. To remove these warnings, create a default report for every navigation node that has been configured.

In general, Scrutiny runs quickly (a few minutes for large EpiCenters), especially when EpiMart is not selected.

FIGURE 100: SCRUTINY DEBUGGING TOOL



☑ Check EpiMeta

☑ Check EpiMart ● A Tables ○ B Tables ● X Tables ○ Y Tables

☐ Prompt before running each check

☐ Quiet mode

Logging Dir: C:\Program Files\Epiphany\4.0\

When Scrutiny runs, it displays a text screen of the checks that are running. If an error is detected, it describes the nature of the error and either how to fix it yourself (usually in EpiCenter Manager) or proposes a solution for you. If you would like it to execute its proposed solution, press **y**. After Scrutiny has executed, it asks if you want to re-run that section, to ensure that the fix was applied successfully.

Warning: In some cases Scrutiny fixes are last resorts, and should be applied only if all else fails. These checks are identified as such, but be sure to read the descriptions carefully before applying any fix.

CONFIGURING WEB PAGES AND TOPICS

A user interacts with the E.piphany system by means of *Web pages*, which provide the user interface for your application. Web pages allow users to generate specific reports or perform specific actions. They are grouped into related sets of activities using *topics*. A *topic* consists of a set of Web pages and the links that you create between those pages.

Links within the E.piphany system perform the following functions, depending on how they are configured.

- They allow users to move from one Web page to another.
- They can carry report information (state) from one Web page to another. This state information includes the filters, attributes, options, and measures (FOAM) that a user has selected.
- They can automatically invoke the report-generation action of the destination Web page.

The combination of state information and report-invocation options associated with a link is referred to as the *behavior* of that link.

Topics are typically derived from templates called topic masters. Depending on the E.piphany solutions that you have purchased, you may have access to one or more topic masters.

The sections that follow discuss how to:

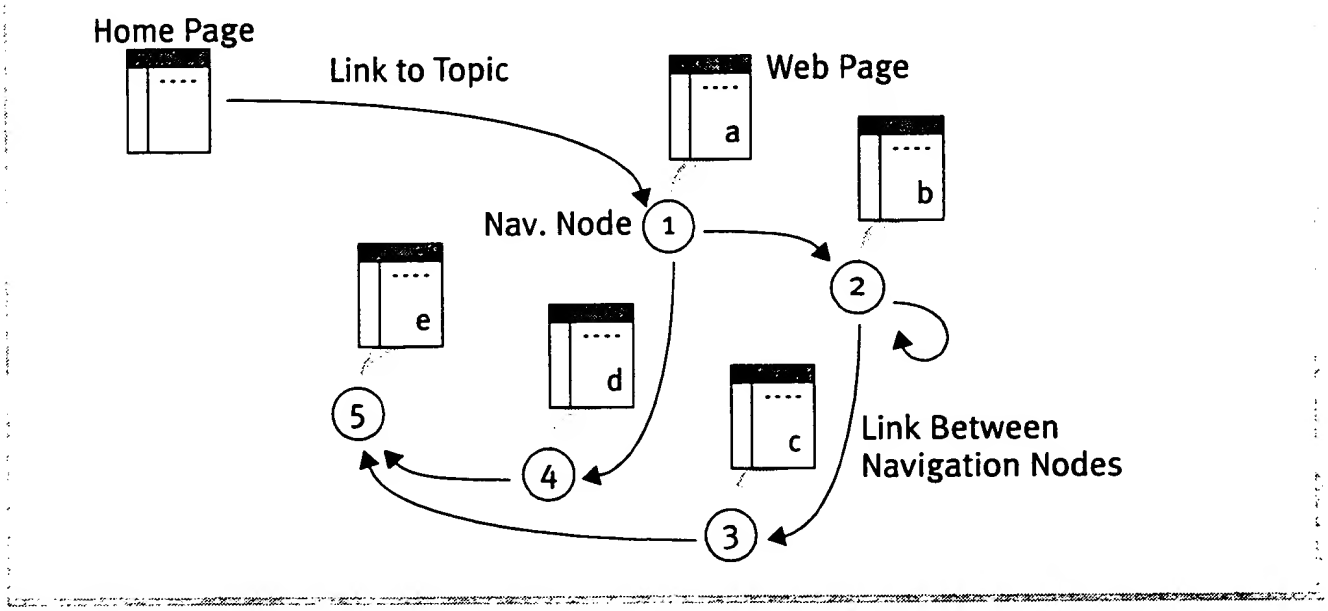
- Design a topic based on the user activities you want to provide
- Create the Web pages that perform each specific activity
- Build a topic from an existing topic master
- Assign Web pages to your topic
- Specify links between Web pages within a topic

- Add links to Web pages in other topics
- Provide site-specific on-line help for your topic and Web pages
- Create new topic masters

DESIGNING A TOPIC

The process of designing a topic is similar to planning an itinerary. You create a map of destinations called ***navigation nodes***, the Web pages that are assigned to each node, and the routes or ***links*** between those node. Figure 101 shows the site map for a sample topic. A navigation node defines a location at which a user stops to perform an activity. The Web page that you assign to a navigation node determines the type of activity that the user can perform at that location.

FIGURE 101: SAMPLE TOPIC MAP



Next, you list the activities that you want your users to perform at each Web page, and the information that needs to be carried from one Web page to another along the way. Table 102, on page 305 shows a sample itinerary for a new topic.

FIGURE 102: SAMPLE ITINERARY FOR A NEW TOPIC

Nav. Node	Web Page	User Activity	Destination of Link	Linking Behavior at Destination
n/a	(Home Page)	Allow user to choose a topic	1	Clear state, display only
1	a	Report sales by region and quarter	2	Carry state, display only
			4	Carry state, execute report
2	b	Drill down (repeatedly) by product, salesperson, customer demographics, and so on	2	Carry state, execute report
			3	Carry state, execute report
3	c	Identify influential attributes	5	Carry state, display only
4	d	Create profiling charts	5	Carry state, display only
5	e	Show highest and lowest values	n/a	n/a

When you have mapped out a primary route, you can consider alternate routes, shortcuts, and side trips. The primary route typically includes the most comprehensive or complicated set of activities that you want your topic to support. The alternate routes support other related activities that the topic can support.

MAPPING USER ACTIVITIES TO WEB PAGES

The reports and data that your topic provides are determined by the Web pages that you configure. Each Web page has a specific type, which determines the reports, lists, or campaigns that users can create. The combination of filters, attributes, options, and measures (FOAM) that you configure for a particular Web page determines the range of data that your users can access through that page.

After you have configured a Web page, you can reuse it in other topics. If you need to create similar reports that give access to different data, you can create different Web pages of the same type. When you create more than one Web page of a given type, E.piphany suggests that you provide mnemonic names to those Web pages to indicate the scope of the data that each page can access.

Table 10 lists the Web page types that E.piphany provides for generating reports, lists, and campaigns. The Web pages that you configure for your users depend on the solutions that you have purchased as part of your E.piphany application.

TABLE 10: WEB PAGE TYPES FOR REPORTS, LISTS, AND CAMPAIGNS (1 OF 2)

Web Page Type	Description
Advanced Rows and Columns	Display measures with respect to a pair of attributes and allow selection of multiple rows for drill-down reports
Basic Rows and Columns	Display measures with respect to a pair of attributes
Calendar	Display calendar of scheduled campaigns
Community Clusters	Identify groupings of values along a set of attributes
Cumulative Projections	Project cumulative results for a set time period based on previous time periods and current results so far
Group Campaign Segment to List Manager	Modify a group campaign segment with Group List Manager
Group Campaign to List Manager	Modify selection criteria for a group campaign with the Group List Manager

TABLE 10: WEB PAGE TYPES FOR REPORTS, LISTS, AND CAMPAIGNS (2 OF 2)

Web Page Type	Description
Group Campaigns	Create campaigns directed toward members of groups
Group List Manager	Create lists based on group demographics
High/Low Clusters	Find combinations of attributes associated with particularly high or low values.
Individual Campaign Segment to List Manager	Modify an individual campaign segment with Individual List Manager
Individual Campaign to List Manager	Modify selection criteria for an individual campaign with Individual List Manager
Individual Campaigns	Create campaigns directed toward individuals
Individual List Manager	Create lists based on individual demographics and transactions
Influences	Identify predictive relationships among a set of attributes
Lifecycles	Project the life cycle for a new product based on the performance of previous like products
Modeling	Identify predictive relationships for use in creating scored lists
Profiling	Chart values along an attribute, or compare values with different filters in effect
Schedule	Add a campaign to the calendar of scheduled campaigns
Scoring	Apply an ordering to a list to rank list members
Trends	Identify trends along a set of attributes and predict future values based on those trends
View List	View list members and download selected attributes
View List from Segment	View list members from within a campaign segment

Table 12 lists Web page types that are used to supply additional state information when users make a transition from certain Web page types to others. Web pages of the following types appear as intermediate nodes between an origination node and a destination node that displays a Web page that creates a report, list, or campaign.

Table 11: Intermediate Web Page Types

Web Page Type	Description
Add a Filter Setting	This Web page type allows users to choose a set of filters to carry into the Web page that is next in the navigation path.
Add FOAM to ...	These Web page types allow users to choose specific filters, options, attributes or measures to carry into the next Web page of a given type.
Add Parameters to Trends	This Web page type allows users to choose parameters to carry into a Trends Web page.
Add Transaction Filters	This Web page type allows users to choose a set of transaction filters to carry into a List Manager or Campaign Manager Web page.
Select a Navigation Path ...	This Web page type displays navigation options without running a report. Path selector Web pages do not carry state. If you want to carry from one Web page to another, you must create a direct link from the source Web page to the destination Web page.
Top N Scores	Set a maximum size for a list to be scored by Scoring

Table 12 lists additional Web page types that perform other functions as part of your E.piphany application.

Table 12: Other Web Page Types

Web Page Type	Description
Home Page	This is the home page for an E.piphany application.
Login Page	This is the login page by which users gain access to an E.piphany application. The login page is provided automatically. It does not appear in the list of Web page types in EpiCenter Manager.
Report Gallery	This is the repository for saved reports, lists, and campaigns.

Creating Web Pages for User Activities

This section describes the process that you typically follow to create Web pages for the user activities that your topic is intended to support. Consult the next section, "Considerations for Specific Web Page Types," on page 311, for information about creating specific Web pages.

1. In the **Presentation** folder of your EpiCenter, right-click the **Web Pages** icon, then choose **New Web Page.**
2. In the General tab of the Web Page dialog box:
   a) Enter the name of your Web page.
   b) Choose a Web page type from the **Web Page Type** drop-down menu
   c) Enter a description of the Web page in the **Description** text box.

   When you choose a Web page type, the contents of the dialog box adjust to allow you to enter information that is appropriate for that type.

3. If the Attributes tab appears, choose the attributes that you want to include in your Web page. Display-area buttons for the row- and column-attribute lists appear at the top of the **Attributes** pane. For example, in a Basic Rows & Columns Web page, the following display-area buttons appear: **Columns, Rows** and **Filters.**

   a) Click on the name of a display area to select it. The attributes that are available appear in the **Object Gallery** pane, and you can drag items from the **Object Gallery** pane to the **Attributes** pane to add them to the display area you have selected.

   b) If you have a large number of attributes for users to select from, you can use categories to reduce the size of the display area on the Web page. Click the **Category** button and choose **Add** from the drop-down menu. Enter the name of a new category, then drag attributes and drop them directly on the appropriate category name that appears in the **Attributes** pane. When the Web page is displayed, only the attributes that fall within the category that a user selects are shown. The **Category** button also allows you to remove or rename categories.

   c) You can adjust the placement of attribute labels within the display area by clicking the **Sort, Up**, or **Down** buttons. You can adjust the indent level for an attribute label on the Web page by clicking the left-arrow or right-arrow button. To activate these buttons, choose an attribute in the **Attributes** pane.

4. If either the Measure Layout or the Transaction Filters tab appears, follow the same process that you used in the previous step to add measures or transaction filters to your Web page.

Note: *EpiCenter Manager allows you to include measures in a Web page that do not necessarily join to attributes that appear on that page. To ensure that end users can obtain meaningful results from a Web page, be sure that the measures you select bear a useful relationship to the attributes that you have already selected.*

5. If the Measure Sets tab appears, drag measures from the upper section of the **Object Gallery** to the **Measure Sets** pane. If you are configuring an Influences Web page, drag target attributes that you want to associate with a measure set from the lower section of the **Object Gallery** to the rows for the applicable measures.

When you have completed these steps, an icon for your new Web page appears in the Web pages folder in EpiCenter Manager.

If you have several Web pages of the same type to create, you can use an existing Web page as a starting template by following these steps:

1. Right-click the icon for the first Web page of that type and then choose **Duplicate** from the pop-up menu.
2. Enter the name of the new Web page in the **Duplicate: Web Page** dialog box.
3. Right-click the icon for the new Web page and choose **Edit** from the pop-up menu.
4. Delete attributes, filters, measures, and other settings that do not apply to the new Web page, and enter the information for those that do.

CONSIDERATIONS FOR SPECIFIC WEB PAGE TYPES

The sections that follow describe the basic functionality and configuration considerations for specific types of Web pages. If a Web page type is not listed, then the directions in the previous section should be sufficient for configuring a Web page of that type. For detailed information about each of the Web page types described in this chapter, refer to the on-line help pages in the following directory:

```
C:\Program Files\Epiphany\instance\Web\help
```

The help pages are in HTML format, and can be viewed with most Web browsers.

THE HOME PAGE

The Home page is configured automatically when you initialize your EpiCenter. It is updated automatically whenever you add a topic. The appearance of the home page for a particular user depends on the topics, favorite reports, and charts to which that user has access, as well as the User Preferences that she or he has set.

You can customize the home page as follows:

- By replacing the logo that appears in the upper-right-hand corner (applies to all Web pages, not just the home page)
- By adding a news banner that appears at the top of the page

Users can customize the page by specifying the number of favorite reports and charts to display and by saving favorite reports and charts. For more information about favorite reports and charts, refer to "Report Gallery," on page 287.

REPLACING THE LOGO

The application logo appears in the upper-right-hand corner of every Web page (excluding pop-up-dialogs). To replace the logo with the logo for your company or solution:

1. Locate a GIF file that contains the logo you want to use.
2. Save a copy of the file called **clientlogo.gif** in the following directory:

   ```
   C:\Program Files\Epiphany\instance\WWWRoot\images
   ```

3. Replace the **clientlogo.gif** file with the GIF file that contains the logo you want to use.

CUSTOMIZING THE NEWS BANNER

The Home page includes a banner that you can customize to display timely information to users. This information can include a message of the day, alerts about changes in the system, and the status of data loads. The news banner appears near the top of the page, above the list of topics and favorite reports.

To customize the news banner, edit the **homepage_news.template** file in the following directory.

```
C:\Program Files\Epiphany\instance\Web\Templates
```

By default, this file contains customization instructions, which E.piphany recommends that you replace.

The **homepage_news.template** file can include valid HTML, including applets. You can also include E.piphany AppServer macros (See Appendix A, "E.piphany Macros.") A particularly useful macro for the banner file is:

```
<!--EP CLASS="com.epiphany.presentation.DateMethod" language="en"
-->
```

which expands to become the date and time of the last successful extraction.

Each E.piphany solution includes a registered copy of the Category Ticker applet by Black Lab Design®, which is an attractive and convenient tool for posting news.

Documentation for this applet is available in the **CategoryTickerDocs.html** file in following directory, and at http://www.javapplets.com.

```
C:\Program Files\Epiphany\instance\WWWRoot\help
```

The **homepage_news.template** file is parsed by the AppServer, not by the dynamic parser of the Web server, so the file must not include ASP scripts. You must use valid HTML syntax, which can include links to other dynamic pages. The contents of this file are included in the body of the Home page, so the file must *not* contain tags that belong outside of the body section of an HTML page. Avoid such tags as `<HTML>`, `<HEAD>`, `<TITLE>`, and `<BODY>`, along with the corresponding closing tags.

A sample **homepage_news.template** is included in the template directory. It uses the Category Ticker applet. If other applets are used, they should be placed in the **WWWRoot** directory.

THE LOGIN PAGE

The Login page is configured automatically when you initialize your EpiCenter. As with the Home page, you can include a banner that displays timely information.

To add a news banner, create or edit the **login_news.template** file in the following directory:

```
C:\Program Files\Epiphany\instance\Web\Templates
```

By default, this file contains customization instructions, which E.piphany recommends that you replace.

This file can include E.piphany AppServer macros, valid HTML body tags, and applets such as the Category Ticker applet, as described in the previous section.

Tip: *E.piphany suggests that you provide instructions for logging in on the Login Web page. If a user must qualify her or his user name with the name of a particular domain in order to gain access to your E.piphany application, a note to this effect can be very helpful.*

THE REPORT GALLERY

The Report Gallery is configured automatically when you initialize your EpiCenter. Refer to “Report Gallery,” on page 287 for details about managing saved reports with the Report Gallery. You can restrict the types of reports that appear in the Report Gallery for a given topic. Refer to “Configuring Navigation Nodes,” on page 337 for details.

INTERMEDIATE WEB PAGE TYPES

The intermediate Web page types listed in Table 11, on page 308 allow users to add state information to carry into a destination Web page. This information can include filters, options, attributes, or measures (FOAM), or other settings that affect the information that appears in a report.

These Web page types are available only in the Navigation Nodes dialog box. You do not configure new Web page contents for these Web page types. Instead, you choose the same Web page as the one that has been assigned to the navigation node into which the added state is to be carried. For instance, if you want to give users the option to add new filters when going from a Basic Rows and Columns or Advanced Rows and Columns Web page to an Influences Web page, you choose the same Influences Web page that is to receive that information as the Web content for the navigation node.

Intermediate Web page types are used between nodes when:

- the Web page at an origin node does not capture all of the information that is necessary to generate a report on the Web page that resides at the destination of a link.
- you want the report at the destination page to be generated automatically (the link to the destination node has the Create Report behavior).

For more information about when to use these Web page types, refer to "Configuring Intermediate Nodes," on page 346.

SELECT A NAVIGATION PATH

The Select a Navigation Path Web page type derives its distinguishing properties from the navigation nodes to which it is assigned. You can create just one Web page for this type and reuse it throughout your E.piphany solution. Creating this Web page involves assigning a label, providing a description, and selecting the Web page type. This Web page type includes no filters, attributes, options, measurements, or other data elements.

BASIC AND ADVANCED ROWS AND COLUMNS

The Rows and Columns Web page types allow users to browse through the information that your datamart contains. Measures are broken out across a pair of attributes that the user selects. Values for one attribute appear as row headings in the report. Values for the other attribute appear as column headings. The measure values that correspond to the intersection of a pair of row and column attribute values appears in each table cell.

The Advanced Rows and Columns Web page type allows users to select multiple attributes to create a multiple-level, indented, drill-down report in a single step. The Basic Rows and Columns Web page type can produce the same report by adding attributes in successive drill-down reports.

It is possible for attributes and measurements that are unrelated to appear on the same Web page. E.piphany recommends that you take care to ensure that the attributes and measures you configure are relevant to each other. For details on configuring a Web page of this type, refer to "Creating Web Pages for User Activities," on page 309.

COMMUNITY CLUSTERS, INFLUENCES, AND MODELING

The Community Clusters Web page allows users to identify clusters of records with similar characteristics. For example, you can use this Web page to find groups of customers who have similar demographic information. The Influences Web page allows users to identify attributes that have important impacts on a measure or another attribute. The Modeling Web page allows users to build predictive models based on predefined targets.

These Web pages use sophisticated decision-tree methods to analyze data. Depending on the size (cardinality) and number of attributes that users choose, the creation of these models can result in slow response times. You might need to consider limiting the number of attributes that you include in these Web pages.

When you configure any of these Web pages, you must identify the primary dimension by which to evaluate your data. For example, if your users want to analyze relationships or rankings among individuals, then the primary dimension is the individual. Similarly, groups might be another good candidate for a primary dimension.

To specify the primary dimension role, select the **Primary** button under the Attributes tab, then choose an attribute that encompasses an entire dimension, or a list-membership attribute, from the **Object Gallery** pane.

Next, you need to determine what attributes might be useful in analyzing this primary dimension. For instance, demographic attributes, such as age, income, or education level, may be useful attributes. The attributes that you include in must either come from the primary-dimension table or stand in a one-to-many relationship with that table.

Warning: When configuring a Community Clusters, Influences, or Modeling Web page that includes source attributes which are not derived from the primary dimension, you must ensure that the one-to-many relationship between source attributes and the primary dimension holds. EpiCenter Manager does not enforce the requirement of a one-to-many relationship, but if this type of relationship does not exist, then some or all queries attempted by users from these Web pages can fail.

To add attributes, click the **Source Attributes** button and then drag and drop attributes from the **Object Gallery** pane. Restrictions on the types of attributes that you can include appear at the bottom of the **Object Gallery** pane.

Finally, you need to specify a Measure Set to be used.

For a Community Clusters Web page, choose a Measure Set of type Clustering that contains a single COUNT measure that counts the number of distinct entities (e.g., individuals) in the primary dimension table. For example, if the primary dimension is individuals, then the measure set must contain a single COUNT measure counts of the number of individuals.

For an Influences or Modeling Web page, the Measure Set can be of type Regression or Classification. When you use a Regression Measure Set, you must specify a COUNT and a SUM role. You can also specify an optional SUMSQUARED role.

If you use a Classification measure, you must also specify a target attribute to use in conjunction with that measure. The target attribute is the attribute that the Influences or Modeling Web page attempts to predict. To select a target attribute, drag it from the **Object Gallery** pane and drop it in the **Target Attribute** folder that appears directly to the right of the Classification measure set that you selected earlier.

If you would like to analyze relationships that are not contained in the primary dimension, such as clusters of individuals based on whether or not they have bought products X, Y, or Z, you can work around the restriction that attributes can come only from the primary dimension.

To do so, you can create supplemental attributes in the individual table that are populated at extraction time and indicate which individuals bought these products. Note however, that these attributes are not automatically synchronized with respect to other dimension values. As dimension values are updated, these attributes can be rendered inconsistent unless you take steps to refresh them from time to time.

Community Clusters, Influences, and Modeling Web pages perform complex statistical analyses that can take a long time to run. The actual running time depends on the number of attributes selected by the user on the Web page, the number of values that these attributes have, and the number of rows in the primary dimension.

As a result, when configuring Web pages of either type, E.piphany suggests that you omit unneeded attributes that have a high cardinality, since these attributes make the application take longer to run and generally have little impact on the results of a report.

Using Lists as Attributes in Community Clusters, Influences, or Modeling Web Pages

When the primary dimension of a Community Clusters, Influences, or Modeling Web page is the same as the individual or group dimension, your Web page can use list membership as an attribute that indicates the group in which an individual is a member.

Slowly Changing Dimensions

When the primary dimension of a Community Clusters, Influences, or Modeling Web page uses the Slowly Changing Dimensions semantic type, a single logical member of the dimension (such as a single individual) may appear as multiple rows in the dimension table if the attribute values of that dimension member have changed over time (for example, an individual got married or moved to a different state).

In this case, Influences and Modeling treat each row in the dimension table as a different member of the dimension when building a model. This may occasionally lead to minor differences in counts between these Web pages and the List Manager Web pages, but should not present any major difficulties.

If you are configuring a Web page that includes measures that perform COUNT DISTINCT operations on fact tables, the dimensions that query those measures can report duplicate values when they contain slowly changing dimensions. To avoid this problem, you can add a column to each of those dimensions that contains a duplicate of the **sskey** value and configure measures that perform COUNT DISTINCT operations on those columns.

Aggregates

If you build an aggregate on the primary dimension of a Community Clusters, Influences, or Modeling Web page that includes all columns of the base dimension (or at least all columns that are used as attributes or filters on the Web page), queries started by that Web page almost always use the aggregate.

PERFORMANCE ISSUES

Two kinds of queries cause Community Clusters, Influences, and Modeling Web pages to adopt complicated query plans: queries involving membership in lists as source attributes or targets, and regression queries in Influences and Modeling that are filtered on dimensions other than the primary dimension. All other things being equal, these kinds of queries tend to take somewhat longer than ordinary queries. If the processing time for queries is a concern to your installation, you may want to avoid these two types of queries.

CUMULATIVE PROJECTIONS

Cumulative Projections Web pages allow users to project results for the current time period (typically a quarter), based on results that have been recorded so far within the period and the pattern of results from previous periods. In Cumulative Projections Web pages, you must assign only relative-date attributes to the Rows display area, such as Days until End of Fiscal Quarter, or Weeks until End of Fiscal Quarter, or Days until End of Month.

GROUP OR INDIVIDUAL CAMPAIGN MANAGER, CALENDAR, AND SCHEDULE

Campaign Manager Web pages allow users to create campaigns based on individual or group attributes. The prerequisites that apply to Group List Manager and Individual List manager also apply to Campaign Manager Web pages. Refer to "Prerequisites for Creating List-Manager Web Pages," on page 325 for information about these prerequisites.

E.piphany recommends that you configure at least one list-management Web page before you configure a Campaign Manager Web page. See "List Manager Web Pages," on page 323 for details.

In addition to the prerequisites for list-management Web pages, Campaign Manager Web pages have the following requirements:

- The **Campaign** and **Cell** base dimensions must be available.
- The **campaign** and **cell** dimension roles must be available.
- You must create attributes for each of the columns in the **campaign** and **cell** dimension roles.

These base dimensions and dimension roles are created automatically when you initialize an EpiCenter with **Include List Manager** checked in the Initialize EpiCenter dialog box.

To create a Campaign Manager Web page:

1. Open the Web Page dialog box by right-clicking the Web Pages icon and choosing New Web Page from the pop-up menu.
2. Fill in the General tab. Choose Group Campaigns or Individual Campaigns as the Web page type.
3. In the Attributes tab, click the **Campaign** button to select the **Campaign** display area. Drag all attributes in the **campaign** and **cell** dimension roles into the **Attributes** list box.
4. Ignore the **Preview** button. Attributes in the **Preview** display area are ignored by Campaign Manager Web pages.
5. Configure filters on attributes of the **group** and **indiv** dimension roles by clicking the **Filter** button and dragging and dropping filters from the Object Gallery. Filters on these attributes are termed demographic filters. Filters on other dimension-role columns are suppressed at run time by the AppServer.
6. Ignore the Transaction Filters tab. Transaction filters are used only in list-management Web pages.

You can customize the labels that the Web page displays for the various factors that are used to calculate the costs and expected returns for a campaign. To do so:

1. Open the **Solutions** folder, then the **Options** folder, and then the **SegmentationCalcs** option.
2. Enter a glossary entry and description for this option in the General tab, then open the Values tab.
3. In the Values tab, choose a value from the **Values** list box and then enter a new label for that value in the **Label** text box.
4. Repeat the previous step for each value in the list.

HIGH/LOW CLUSTERS

High/Low Clusters Web pages allow users to identify the highest and lowest values for a set of attributes with respect to a measure. These Web pages display only one list of attributes, rather than the pair of attribute-lists that appear in the Rows and Columns Web pages. The Attributes tab for a High/Low Clusters Web page includes the **DimSingle** display-area button and **Filters** display-area button.

LIFECYCLES

Life Cycles Web pages allow users to project future values based on patterns established by previous long-term cycles. For instance, if you have sales records for previous products, your users can use the performance figures for those products over time as a model for predicting the performance of a new product that they plan to introduce.

You must assign only absolute date attributes to the Columns display area. You can assign attributes of any type to the Rows display area.

Note: *The columns in Lifecycle Web pages should be absolute date columns, such as the fiscal year or the month of a particular year, such as January 1998. They should not be relative or cyclical columns such as month (that is, January, February, and so forth).*

LIST MANAGER WEB PAGES

The following Web page types allow users to derive lists of groups or individuals, such as customers, resellers, households, contacts, or other similar entities:

- Group List Manager
- Group Campaign to List Manager
- Group Campaign Segment to List Manager
- Individual List Manager
- Individual Campaign to List Manager
- Individual Campaign Segment to List Manager

Users can generate lists based on demographic data and transaction histories. Demographic data includes attribute data that appears in your Group or Individual attribute tables. Transaction histories are derived from fact tables.

Unlike most data-warehousing applications, list-management applications use fact data (transactions) to classify information about customers, prospects, and so on. (In most other cases, attribute data is used to classify facts, such as by breaking out sales data by customer.) List members can be selected or excluded based on their participation in a particular type of transaction.

The list-manager Web Page types provide sophisticated list-management capabilities, such as:

- Demographic filtering based on attributes of the **group** or **indiv** dimension roles
- Transaction filtering based on measures and attributes of other dimension roles
- Generation of approximate and exact member counts

The **group** and **indiv** dimension roles are provided for you when you configure your EpiCenter for lists and campaigns (if you ensure that the **Include List Manager** check box of the Initialize EpiCenter dialog box is checked). These dimension roles provide attributes that users can employ to break out lists for various purposes.

Demographic Filters

A user of a list-manager Web page applies filters to restrict list membership based on attributes of the **group** or **indiv** dimension roles. Filters based on these attributes are called *demographic filters*. After the user selects a filter in a List Manager Web page, he or she can find out how many matches it corresponds to in the database. These filters are configured in the same manner as filters in other Web pages.

TRANSACTION FILTERS

You can configure list-manager Web pages to allow users to filter individual and group dimensions based on participation in transactions. A *transaction* is an action that an individual or a member of a group might participate in, such a purchase, a return, a call to a call center, the receipt of a treatment, or a response to a campaign. Users can query against measures along these types of transactions. For instance, they might want to know how much a customer has purchased, how many campaigns a prospect has received, or how often the members of a household place calls to a call center. They might also want to know which customers have *not* participated in a certain type of transaction.

To allow users to filter on transaction data in this manner, you need to define *transaction filters* in addition to regular filters. Refer to "Configuring List-Manager Web Pages," on page 328 for information about configuring transaction filters.

PREREQUISITES FOR CREATING LIST-MANAGER WEB PAGES

If you want to provide E.piphany list-management and campaign-management capabilities to your users, you must be sure that your EpiCenter has been configured with the built-in fact tables, dimension tables, and dimension roles that support these capabilities. These structures are only built when the **Include List Manager** check box is checked in the Initialize EpiCenter dialog box at the time you initialize your EpiCenter. To verify that these supporting structures are in place:

1. Open the **Schema** folder and then the **Base Dimensions** folder, then look for the **GroupDim** and **IndiviDim** base dimensions.
2. Open the **Dimension Roles** folder and look for the **group** and **indiv** dimension roles.
3. Open the **Facts** folder and look for the **Communication** and **Ind_Group_Joiner** fact tables.

If these tables are not present, you must initialize a new EpiCenter. In order to preserve your work so far, export your metadata. Refer to Appendix G, "Export/Import of Metadata," for information on how to import this metadata into a new EpiCenter.

Make sure that you have defined extraction jobs to populate these tables.

ENABLING APPROXIMATE COUNTS

The list-manager Web pages provide a quick-count feature that uses sampling to estimate the counts for subsets of lists that fit the selection criteria a user has chosen. These estimates allow the queries that calculate these counts to complete in the time it takes to evaluate the sample as opposed to the entire dimension.

You can enable this feature by assigning a positive value to the **min_sample_invlog10** option in the General tab of the Configuration dialog box (see "General Settings," on page 417). The value that you set in this option corresponds to the inverse log (base 10) of the sampling probability. For example, a value of 2 would create a 1-percent sample ($1/10^2$).

THE IND_GROUP_JOINER FACT TABLE

The *Ind_Group_Joiner* table is a special-purpose fact table that defines the many-to-one relationship between individuals and groups. This table identifies demographic dimensions and assigns individuals to groups. Each of these actions is discussed below.

- Identifying demographic dimensions

  The Momentum Builder program and the Application Server use the **indiv** and **group** dimension roles associated with the **Ind_Group_Joiner** table to identify the individual and group base dimension tables.

- Assigning of individuals to groups

  The data in this fact table determines which individual belongs to which group.

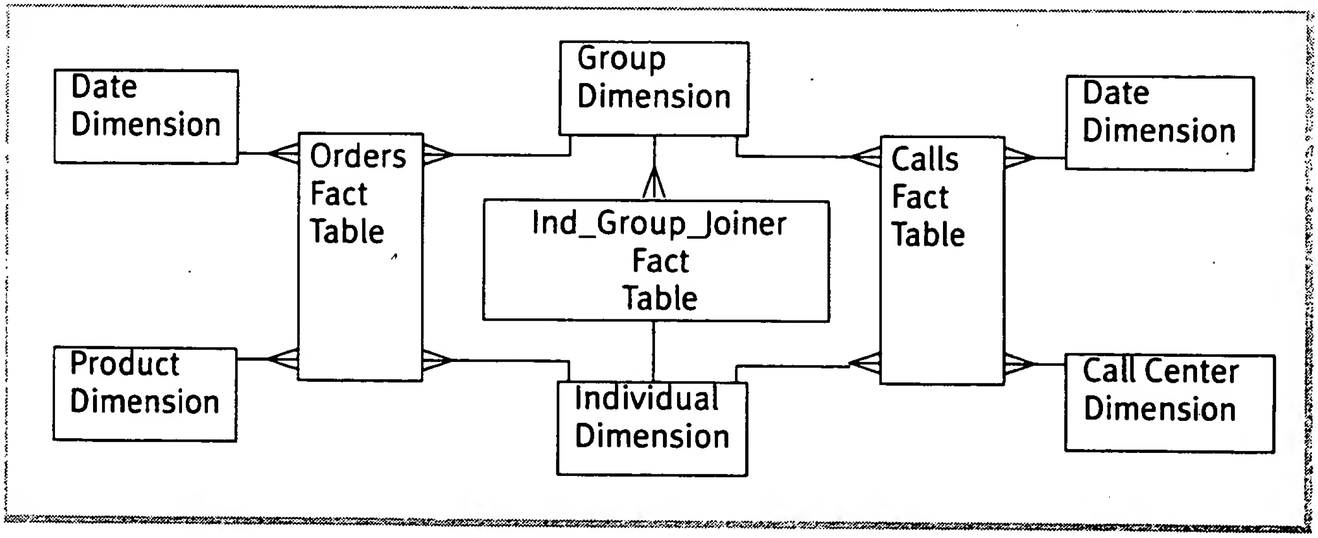
- Defining who appears in lists and campaigns

  The table also defines the individuals and groups that are a part of the selection pool for lists and campaigns. Any individual or group that does not have its key in this table is not available for inclusion in a list or campaign. Even if you have individuals that belong to no group or groups without individuals, you still need to enter their keys in this table to include them in your lists and campaigns.

Table 103, on page 327 illustrates a typical star schema for E.piphany list-generation data. In this example, there are two regular fact tables called **Orders** and **Calls**. There are also dimension tables associated with each of those facts in addition to the **Group** and **Individual** dimension tables. The **Ind_Group_Joiner** table is a third fact table that implements the many-to-one relationship between the **Individual** and **Group** dimensions. With a schema such as the one shown, you could generate lists based on:

- Demographic data in either the group or Individual dimension tables
- The total amount of orders that have been placed by an individual
- The combined total of orders per group (such as a household or company)
- The total number of calls in a given time period for an individual or group placed to a given call center

FIGURE 103: SCHEMA FOR LIST-GENERATION DATA



FACT-TABLE CLUSTERS AND COUNTS

Clusters and counts make transaction filtering faster. A *cluster* is a copy of the fact table sorted on an attribute. A large table should be sorted on disk with its leading term being the most selective filter.

Counts keep statistics about how often entries for a group or individual appear in a fact table. Counts are used by the query engine to select the best-clustered copy of a fact table.

To enhance the performance of queries that use transaction filters, choose at least one cluster for each fact table that contains transaction data that your users filter against. Be aware that clusters consume the same amount of disk space as the fact tables from which they are copied. You must define a cluster in order to define counts for a fact table.

CONFIGURING LIST-MANAGER WEB PAGES

After you enter a name, choose a Web page type, and enter a description, the actions involved in configuring a list-manager Web page include:

- Adding attributes to the Preview display area of the Attributes tab
- Adding demographic filters in the Attributes tab
- Adding transaction filters in the Transaction Filters tab

The sections that follow discuss these actions.

CONFIGURING ATTRIBUTES

You set up attributes for list-management Web pages using the Attributes tab. Restrictions on the types of attributes that you can include appear below the Object Gallery list box. Although the **Campaign** button appears, campaign attributes are unused in list-management Web pages. Preview attributes are required. You configure these attributes by clicking the **Preview** display-area button and then dragging and dropping attributes from the **Object Gallery** pane.

If the Web page type is Group List Manager, Group Campaign to List Manager, or Group Campaign Segment to List Manager, then only group attributes are allowed. If the Web page type is Individual List Manager, Individual Campaign to List Manager, or Individual Campaign Segment to List Manager, the attributes can pertain either to an individual or the group of which that individual is a member.

CONFIGURING DEMOGRAPHIC FILTERS

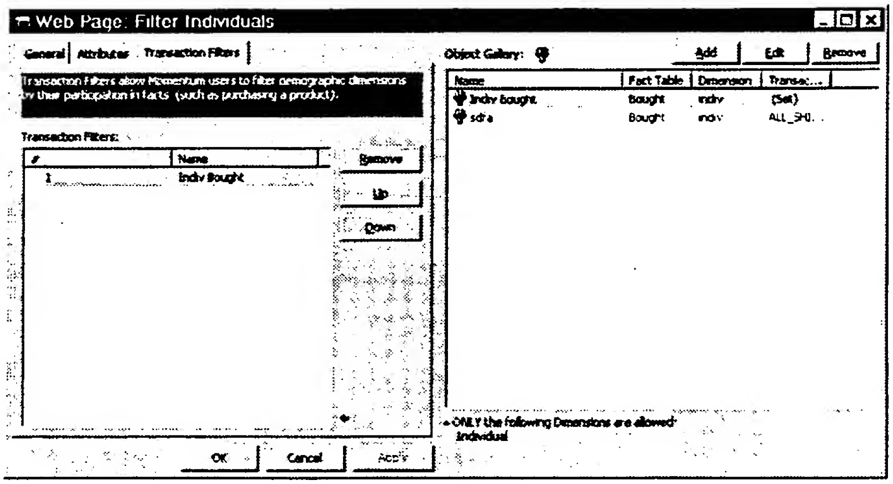
Demographic filters are based on attributes of the **group** and **indiv** dimension roles. In other respects, they are just like the filters used in other Web pages. If you include nondemographic filters in the Filters display area, those filters are suppressed by the AppServer at run time. You can include filters for other types of dimensions when you configure transaction filters.

CONFIGURING TRANSACTION FILTERS

Take the following steps to configure transaction filters for a list-manager Web page:

1. Open the Transaction Filters tab of the Web Page dialog box for a Web page of type Filter Groups or Filter Individuals. Figure 104, on page 330 illustrates this tab. Click the **Add** button to add a new transaction filter. Click **Edit** or double-click a row in the Object Gallery to edit an existing transaction filter.

FIGURE 104: TRANSACTION FILTERS TAB OF THE WEB PAGE DIALOG BOX



Transaction Filters allow Momentum users to filter demographic dimensions by their participation in facts (such as purchasing a product).

Transaction Filters:

#	Name
1	Indiv Bought

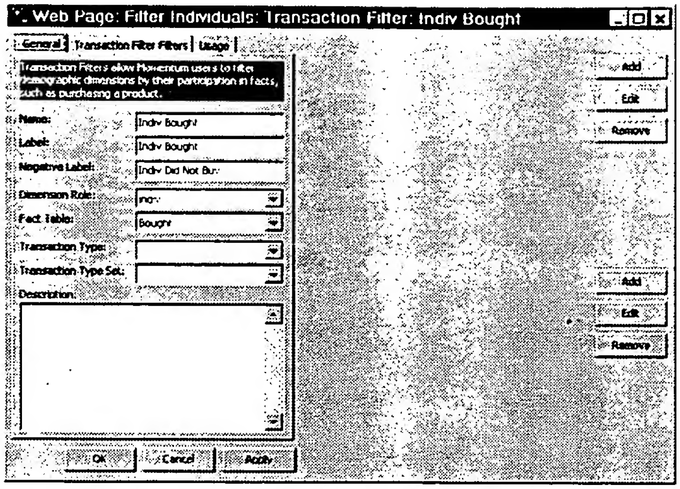
Name	Fact Table	Dimension	Transac...
Indiv bought	Bought	indiv	(Set)
sdfa	Bought	indiv	ALL_SHI...

ONLY the following Dimensions are allowed:
Individual

2. In the General tab of the Transaction Filter dialog box (Figure 105), enter the following information about the transaction filter:
   - The name

     When you enter the name, EpiCenter Manager fills in values for the label and negative label. The negative label is the name of a filter that excludes values that qualify for the transaction filter.
   - The dimension role (either **group** or **indiv**)
   - The transaction type to apply to the transaction filter
   - The transaction type set
   - The description for this transaction filter

FIGURE 105: TRANSACTION FILTER DIALOG BOX



Web Page: Filter Individuals: Transaction Filter: Indiv Bought

Transaction Filters allow Momentum users to filter demographic dimensions by their participation in facts, such as purchasing a product.

Name: Indiv Bought

Label: Indiv Bought

Negative Label: Indiv Did Not Buy

Dimension Role: Indiv

Fact Table: Bought

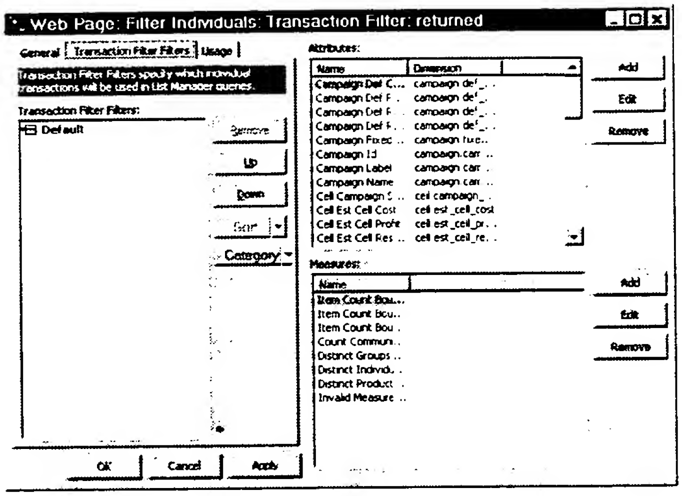
Transaction Type:

Transaction Type Set:

Description:

3. In the Transaction Filter Filters tab (Figure 106, on page 332), drag and drop the attributes and measures that you want to allow in this transaction filter. To add new candidate attributes, click **Add** near the Attributes text box. To add new measures, click **Add** near the Measures text box. To edit an existing attribute or measure, choose it and then click the appropriate Edit button. You can include measures in your transaction filters.

FIGURE 106: TRANSACTION FILTER FILTERS TAB


Web Page: Filter Individuals: Transaction Filter: returned
General
Transaction Filter Filters
Usage
Remove
Up
Down
Sort
Category
Add
Edit
Remove
Add
Edit
Remove
OK
Cancel
Apply

Transaction Filter Filters specify which individual transactions will be used in List Manager queries.

Transaction Filter Filters:

- Default

Attributes:

Name	Dimension
Campaign Def C...	campaign def_..
Campaign Def F..	campaign def_..
Campaign Def F..	campaign def_..
Campaign Def F..	campaign def_..
Campaign Fixed ..	campaign fixe..
Campaign Id	campaign.cam ..
Campaign Label	campaign cam ..
Campaign Name	campaign cam ..
Cell Campaign S ..	cell campaign_ .
Cell Est Cell Cost	cell est _cell_cost
Cell Est Cell Profit	cell est _cell_pr..
Cell Est Cell Res ..	cell est _cell_re..

Measures:

Name
Item Count Bou...
Item Count Bou..
Item Count Bou .
Count Commun..
Distinct Groups ..
Distinct Individ..
Distinct Product .
Invalid Measure ..

4. Follow the directions in Chapter 3, “EpiCenter Manager,” to configure new attributes or measures that you intend to use in the Transaction Filter Attribute dialog box or the Transaction Filter Measure dialog box.

CUSTOMIZING LABELS

You can customize the labels that appear in list-manager Web pages for groups or individuals. To replace the singular and plural labels for groups or individuals:

1. Open the **Solutions** folder, then the **Options** folder, and then open either the **GroupLabels** or the **IndividualLabels** option.
2. Enter a label, a glossary entry, and a description for this option in the General tab of the Option dialog box.
3. In the Values tab, choose a value from the **Values** list box. Enter a new label for that value in the **Label** text box.
4. Repeat the previous step for each value in the list.

PROFILING

The Profiling Web page allows users to create charts and perform comparisons by applying combinations of attributes and filters. Profiling Web pages use only one display area for attributes. These Web pages are not adversely affected by high-cardinality attributes. Users can use a variety of filters to create comparison reports, so E.piphany suggests that you include other filters that might be of use.

SCORING

The Scoring Web page allows users to rank members of a list of groups or individual using a predictive model that they have created with either a Community Clusters, Influences or Modeling Web page. Users also can score lists based on a measure. You only need to create one Scoring web page in your EpiCenter. Refer to "Configuring Navigation Nodes for Scoring Web Pages," on page 349 for information on how the Scoring Web page is used in topics.

TRENDS

The Trends Web page allows users to calculate projections based on trends that are inherent in the data with respect to a set of attributes. The Trends Web page has two display areas, **Columns** and **Rows**. You must assign only absolute time attributes to **Columns**. You can assign any attribute to **Rows**.

VIEW LIST AND VIEW LIST FROM SEGMENT

The View List and View List from Segment Web pages allows users to display and download selected attributes for list members. This page also provides access to other list-related Web pages for scoring, updating selection criteria, and creating campaigns.

TOPICS AND TOPIC MASTERS

A topic master is the template for a topic. E.piphany provides topic masters that embody the solution maps for on-line analytical processing (OLAP), database marketing, and e-commerce applications. The topic masters that you have available depend on the E.piphany solutions that you have purchased.

A topic master specifies a mapping of navigation nodes and links. A navigation node is a location within the topic at which a Web page can reside. Each navigation node in a topic master has an associated Web page type. When you create a topic from a topic master, you assign a Web page of the appropriate type to each navigation node that you choose to include in your topic.

You do not need to assign a Web page to a navigation node if your topic does not require a Web page of that type. You can either remove the navigation node from the topic or leave it unassigned. The AppServer displays links to navigation nodes that do not have Web pages assigned to them, but it disables those links. However, the AppServer does *not* display links to nodes that users do not have permission to visit. You can hide disabled links either by omitting the target nodes of those links from groups to which users have access, or by unchecking the Visible check box associated with each disabled link in the Links tab of the Navigation Node dialog box. Refer to “Assigning Links and Selecting Behaviors,” on page 341 for details.

If a navigation node is inaccessible, links from that node to other nodes are disabled. If all of the navigation nodes with links to a node that your topic needs are disabled, users cannot reach that node. In a situation like this, you can add a link to the node that you require from another node that is still reachable by users.

BUILDING A TOPIC

The process of building a topic includes the following stages:

1. Creating Web pages for the navigation nodes that you intend to use (if you have not done so already)
2. Creating the topic from a topic master
3. Modifying the structure of the topic by adding or removing navigation nodes (optional)
4. Assigning Web pages to navigation nodes (for nodes that do not have Web pages assigned by default)
5. Choosing behaviors for the links between Web pages

You can also create topics from scratch, without using a topic master as a template. However, E.piphany does not recommend that method. Instead, it is often more convenient to use the topic masters that E.piphany has supplied as part of your application.

CREATING A TOPIC FROM A TOPIC MASTER

To create a new topic, take the following steps:

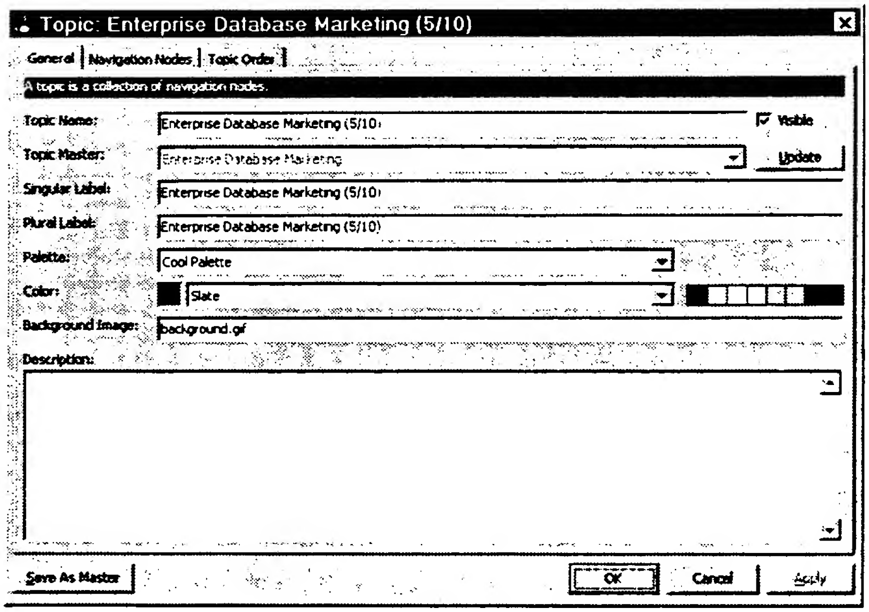
1. Expand the **Presentation** folder for your EpiCenter in EpiCenter Manager, right-click the **Topics** icon, and then choose **New Topic** from the pop-up menu to bring up the Topic dialog box, as shown in Figure 107, on page 336.
2. In the General tab, add the following information:
   a) Enter the name of your new topic in the **Topic Name** text box. As you do, EpiCenter Manager fills in the **Singular** and **Plural Label** text boxes. Edit the contents of these boxes as needed.
   b) Choose a topic master from the **Topic Master** drop-down menu.
   c) Pick a color palette for the topic bar from the **Pallete** drop-down menu. Then pick a color from the **Color** menu.

d) The default background image is suitable for most applications, and E.piphany suggests that you leave the **Background Image** text box set to the default value. Enter a description of the topic including the intended audience and usage.

e) Leave the **Visible** check box checked, unless you do not want this topic to be accessible from the home page.

f) Click the **Apply** button to initialize your topic.

FIGURE 107: THE TOPIC DIALOG BOX



Topic: Enterprise Database Marketing (5/10)

A topic is a collection of navigation nodes.

Topic Name: Enterprise Database Marketing (5/10) ☑ Visible

Topic Master: Enterprise Database Marketing

Singular Label: Enterprise Database Marketing (5/10)

Plural Label: Enterprise Database Marketing (5/10)

Palette: Cool Palette

Color: Slate

Background Image: background.gif

Description:

3. In the Topic Order tab, use the **Up** and **Down** buttons to adjust the placement of the topic bar on the Home page.
4. Click **Apply** to apply the settings you have selected to your topic.

At this point, you have created a topic with a default set of navigation nodes and links.

The **Update** button allows you to refresh the topic from the topic master from which it has been created. When you refresh a topic, some of updates that you have made to that topic are lost.

CONFIGURING NAVIGATION NODES

This section discusses the process for configuring the navigation nodes in a topic, which involves the following activities:

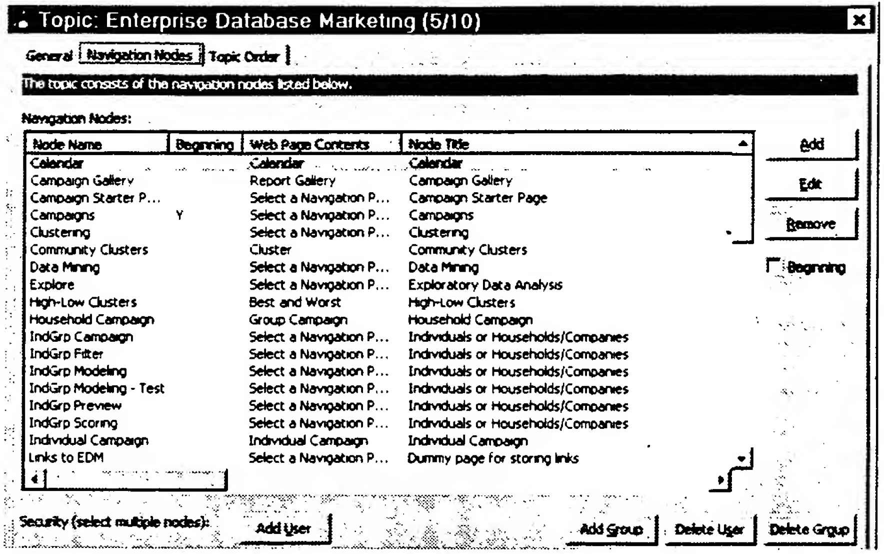
- Assigning Web pages to navigation nodes
- Configuring links and link behaviors
- Providing access to users and groups
- Optionally configuring additional navigation nodes for a topic
- Optionally hiding or deleting unneeded navigation nodes
- Providing on-line help for your topics and Web pages

If you have created only one Web page of a given Web page type, EpiCenter Manager assigns that Web page to each node in your topic that is configured to display a Web page of that type. Default links and link behaviors are inherited from the topic master that you choose. These default settings allow you to complete the process of configuring navigation nodes by making selective adjustments, rather than having to manually indicate every node and link.

The Navigation Nodes tab displays the name, the Web page to be displayed, and the title of each navigation node in a list box. The **Node Title** column shows the text that appears at the top of the Web page that is displayed at a node (Figure 108, on page 338). To reach this tab if it is not currently open, right-click the topic icon in the Presentation folder of your EpiCenter, then choose **Edit** from the pop-up menu, and then click **Navigation Nodes**.

Configuring a navigation node involves assigning Web page contents, granting access to users and groups, choosing Report-Gallery properties, and configuring links. The remainder of this section describes all of these actions except the configuration of links, which is discussed in the next section.

Figure 108: The Navigation Nodes Page of the Topic Dialog Box



The topic consists of the navigation nodes listed below.

Navigation Nodes:

Node Name	Beginning	Web Page Contents	Node Title
Calendar		Calendar	Calendar
Campaign Gallery		Report Gallery	Campaign Gallery
Campaign Starter P...		Select a Navigation P...	Campaign Starter Page
Campaigns	Y	Select a Navigation P...	Campaigns
Clustering		Select a Navigation P...	Clustering
Community Clusters		Cluster	Community Clusters
Data Mining		Select a Navigation P...	Data Mining
Explore		Select a Navigation P...	Exploratory Data Analysis
High-Low Clusters		Best and Worst	High-Low Clusters
Household Campaign		Group Campaign	Household Campaign
IndGrp Campaign		Select a Navigation P...	Individuals or Households/Companies
IndGrp Filter		Select a Navigation P...	Individuals or Households/Companies
IndGrp Modeling		Select a Navigation P...	Individuals or Households/Companies
IndGrp Modeling - Test		Select a Navigation P...	Individuals or Households/Companies
IndGrp Preview		Select a Navigation P...	Individuals or Households/Companies
IndGrp Scoring		Select a Navigation P...	Individuals or Households/Companies
Individual Campaign		Individual Campaign	Individual Campaign
Links to EDM		Select a Navigation P...	Dummy page for storing links

Security (select multiple nodes):

Each topic must have a single beginning node. This node is displayed when a user follows the link to a topic on the Home page. A Y in the **Beginning** column of the **Navigation Nodes** list box indicates the beginning node for the topic. Although you can choose a beginning node by selecting a node from the list box and clicking the **Beginning** check box, E.piphany recommends that you use the beginning node that has been set by default in the topic master.

Assigning Web Pages to Navigation Nodes

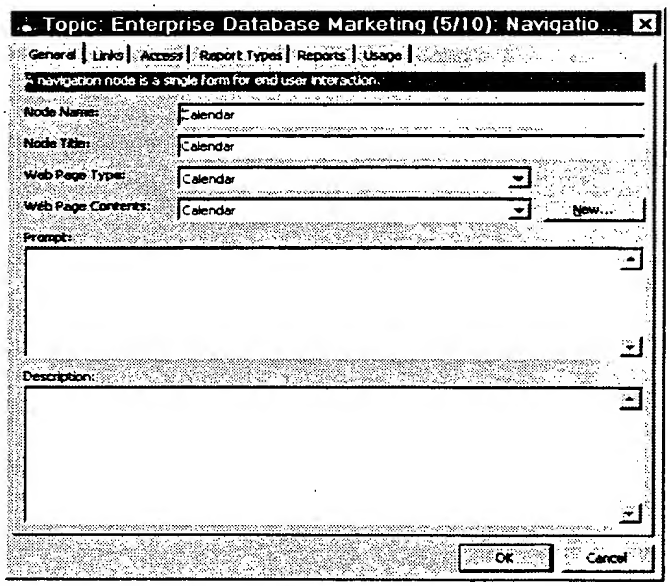
The **Web Page Contents** field of the Navigation Nodes tab displays the name of the Web page that is assigned to a navigation node. If this field is blank, you must either:

- Create a Web page of the appropriate type and then assign it to the node
- Choose from among the several Web pages of that type that are currently available and assign it to the node
- Delete the node if it is unneeded, or assign a different Web page type to that node

To assign a Web page to a navigation node, take the following steps:

1. Double-click the row that displays the node to which you want to assign a Web page, or select that row and click **Edit**. Either action opens the Navigation Node dialog box shown in Figure 109.

FIGURE 109: THE NAVIGATION NODE DIALOG BOX



Topic: Enterprise Database Marketing (5/10): Navigatio...

A navigation node is a single form for end user interaction.

Node Name: Calendar

Node Title: Calendar

Web Page Type: Calendar

Web Page Contents: Calendar

Prompt:

Description:

2. If a Web page that produces the specific report you require does not already exist, click **New** and then follow the instructions in “Creating Web Pages for User Activities,” on page 309 to create it.
3. You can change the Web page type that is assigned to a navigation node by choosing a type from the **Web Page Type** list box. Table 10, on page 306 contains a list of Web page types that generate reports. Table 12, on page 309 contains a list of Web page types that perform other functions. For ease of maintenance, E.piphany suggests that you use discretion in changing the Web page type assigned to a navigation node.
4. Choose a web page from the **Web Page Contents** list box. This list box displays only Web pages that can be used as the content of the chosen Web page type.
5. Enter a prompt that describes the actions that you want users to perform at this node. This prompt appears near the top of the Web page.

6. Enter a description of the node in the **Description** text box.
7. The Access tab displays the groups and users who currently have access to the navigation node. Use the **Add Group** and **Add User** buttons to give access to selected groups and users. To remove access, choose a group or user from the appropriate list box and click the corresponding **Remove** button.
8. The Report Types tab displays the types of report files that the Report Gallery displays when accessed from the current navigation node. Use the **Add Types** button to add report types. Select a report type from the list box and click **Remove** to eliminate it from the display for this node. To restrict reports that the Report Gallery displays to those in the current topic, check **Only show reports from this topic.**
9. The Reports tab lists all reports that have been saved from the current navigation node. The **Remove** buttons allow you to remove a selected report. The **Edit** button allows you to update the properties of a selected report.

CREATING DEFAULT REPORTS

E.piphany recommends that you create a default report for the Web page you configure at each navigation node. You can create a default report for a Web page by:

1. Starting the Application Server. Do this after you have completed configuring your application and run Scrutiny to validate your configuration.
2. Navigating to the node. Your Web browser displays the Web page at that node.
3. Choosing the filters, options, attributes, and measures that shall act as default settings.
4. Clicking the **Save** button to display the Report Gallery

5. Navigating to the **Public/Defaults** folder and saving the report there. The name that you choose does not matter, although a mnemonic name makes it easier to distinguish the default report for one node from another node of the same Web page type.

Note: *When you create a default report for a node that displays a High/Low Clusters Web page, be sure to open the* **compare to other filters / measurement** *pop-up and fill in a default for the comparison measure. E.piphany suggests that you use the same measurement as the default measurement that appears on the main Web page. Set this comparison measurement even if the default report uses the disproportionately or absolutely comparison method, as a user might select compare to other filters/measurements but forget to fill in a comparison measure.*

Assigning Links and Selecting Behaviors

Links provide the means for users to jump from the Web page at one navigation node to the Web page at another node. Each navigation node has a set of links that have been assigned to it by default from the topic master. The Links tab of the Navigation Node dialog box allows you to assign links to navigation nodes. These links specify the destination nodes and behaviors to perform as part of the transition from the current node to a destination node.

Links are assigned to the navigation nodes from which they originate. The navigation nodes that are derived from a topic master include a number of links that are provided by default. In general, you only need to assign links to nodes or adjust the behavior of a link when you:

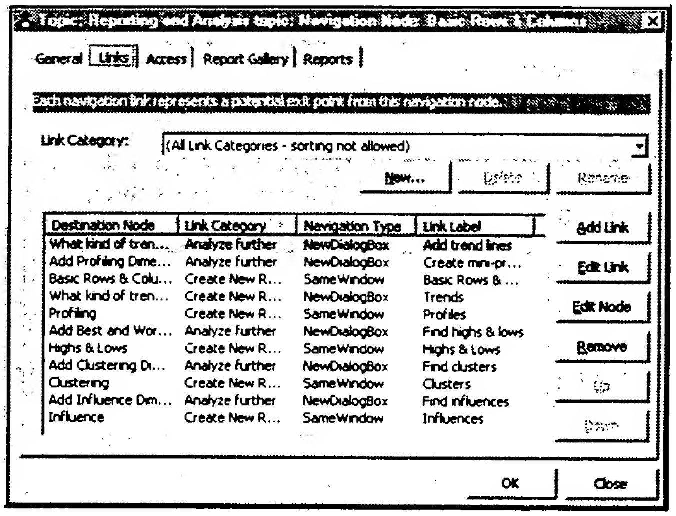
- Add new nodes
- Delete existing nodes
- Provide new, alternative links that appear in addition to the links that are derived from the topic master

If you choose, you can display alternative links in a separate list on the Web page by assigning them to a new link category.

You assign new links, adjust the behavior or destination of existing links, add links to existing categories, or create new categories of links with the Links tab of the Navigation Node dialog box. To reach this tab if it is not already open:

1. Open your topic in the **Topics** folder, which resides in the **Presentation** folder for your EpiCenter in EpiCenter Manager.
2. Open the Navigation Nodes tab.
3. Double-click the row for a node in the Navigation Nodes list box, or select a row and click **Edit**, to display the Navigation Node dialog box.
4. Click the Links tab. Figure 110 shows the contents of this tab.

Figure 110: The Links Tab of the Navigation Node Dialog Box



Topic: Reporting and Analysis topic: Navigation Node: Basic Rows & Columns

Each navigation link represents a potential exit point from this navigation node.

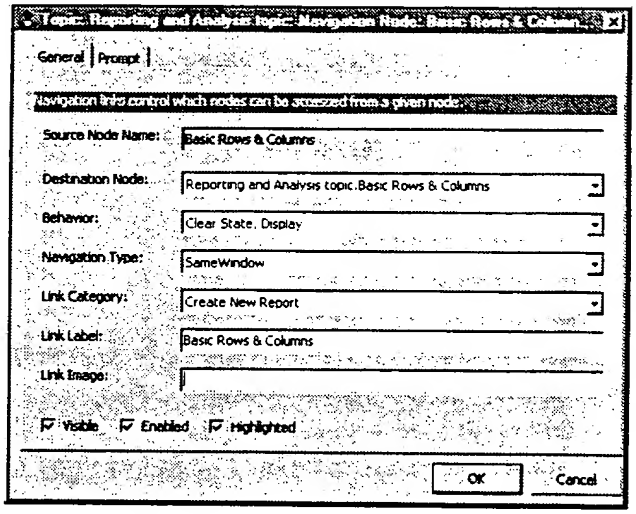
Link Category: (All Link Categories - sorting not allowed)

Destination Node	Link Category	Navigation Type	Link Label
What kind of tren...	Analyze further	NewDialogBox	Add trend lines
Add Profiling Dime...	Analyze further	NewDialogBox	Create mini-pr...
Basic Rows & Colu...	Create New R...	SameWindow	Basic Rows & ...
What kind of tren...	Create New R...	NewDialogBox	Trends
Profiling	Create New R...	SameWindow	Profiles
Add Best and Wor...	Analyze further	NewDialogBox	Find highs & lows
Highs & Lows	Create New R...	SameWindow	Highs & Lows
Add Clustering Di...	Analyze further	NewDialogBox	Find clusters
Clustering	Create New R...	SameWindow	Clusters
Add Influence Dim...	Analyze further	NewDialogBox	Find influences
Influence	Create New R...	SameWindow	Influences

By default, all of the links that have been assigned to a node appear in the list box. You can view the links for a specific link category only by choosing a category from the **Link Category** list box. When you view the links in a specific category, you can rearrange the order in which they appear using the **Up** and **Down** buttons. Links in a category are displayed to the user in the order in which they appear within this dialog box.

You can create a new link category by clicking **New**. You can rename or delete a link category by selecting it from the list box and clicking **Rename** or **Delete**. To add a new link, click **Add Link**. To edit an existing link, select the link and click **Edit Link**. When you add or edit a link, the **Navigation Link** dialog box (Figure 111, on page 343) appears.

FIGURE 111: THE NAVIGATION LINK DIALOG BOX



Topic: Reporting and Analysis topic: Navigation Node: Basic Rows & Column...

Navigation links control which nodes can be accessed from a given node.

Source Node Name: Basic Rows & Columns

Destination Node: Reporting and Analysis topic.Basic Rows & Columns

Behavior: Clear State, Display

Navigation Type: SameWindow

Link Category: Create New Report

Link Label: Basic Rows & Columns

Link Image:

☑ Visible ☑ Enabled ☑ Highlighted

The name of the current node is displayed under **Source Node Name** in the **Navigation Link** dialog box. It cannot be edited. To set the properties of the link:

1. Choose a destination from the **Destination Node** list box. The link leads to the node that you indicate. The destination node can be in any topic, and the list box lists both the topic and node names of all available nodes.
2. Choose a behavior from the Behavior list box. Each behavior includes a pair of components. The first component indicates whether to carry the state information that a user has selected in the current Web page to the Web page at the destination node. The second component indicates whether or not to generate a report when opening the destination Web page.

The behavior components describe the actions that are performed when the link is followed. Table 13 describes the behavior components. The combinations of components that are available for a particular link depend on the type of Web page that resides at the destination node.

TABLE 13: LINK BEHAVIOR COMPONENTS

Behavior Component	Description
Carry State	This behavior component specifies that the state of the current web page (which includes such things as options, filter settings, selected attributes, and lists) should be preserved for use by the destination Web page
Clear State	indicates that the state should not be carried to the destination Web page
Create Report	indicates that a report should be generated (generally by means of a datamart query) for display by the web page at the next node.
Display	indicates that the web page at the next node should simply display information without generating a database query.

3. Choose a navigation type. This indicates the type of window in which the next web page will be displayed. Table 14 describes the available options.

TABLE 14: NAVIGATION TYPES

Navigation Type	Description
Same Window	The destination node is displayed in the same window as the current node.
New Browser Window	The destination node is displayed in a new browser window. In this case, the current node remains in the current browser window.
New Dialog Box	The destination node is displayed in a new pop-up window. The current node remains in the current browser window.
Opener Window	The destination node is displayed in the window from which the current node was opened. For example, if the current node is a popup window, this displays the destination node in the window from which the popup was opened.

4. Choose a link category from the **Link Category** list box or create a new one by clicking the **New** button. Link categories allow you to group links that you want to appear together on the Web page. Links are typically displayed in the left-hand side bar as a tabular list, with the category name shown as the list heading.

   Some link categories are treated as special cases by certain Web pages. These special-purpose categories are described in Table 15. If a Web page does not recognize a link category as a special case, it displays the links in that category in the side bar, in the same fashion as any other link category.

TABLE 15: SPECIAL-PURPOSE LINK CATEGORIES

Link Category	Web Pages	Placement and Usage
Analyze Further	Basic Rows and Columns, Cluster, Influences, Lifecycles, Profiling, Quarter Projections, Scoring, and Trends	Links appear in the results section of the Web page only.
Create New Report	All	Links appear in the report section of the Web page only.
Multiple Choices	Select a Navigation Path	Links appear in the body of the Web page.
Wizard Steps	All	Links appear at the bottom of the report section.
You Are Here	All	Links appear in the report section of the Web page only.

5. Enter a label in the **Link Label** text box. This text appears as the link.
6. Check the **Visible** check box if you want to have the link visible to the user.
7. Check the **Enabled** check box to enable the link. If **Visible** is checked and **Enabled** is unchecked, then the user will see the link label but will not be able to follow the link.
8. Check **Highlighted** to display the link with the background color of the topic.

9. For links in the Multiple Choice and Wizard Steps categories, you can enter the name of an image in the **Link Image** text box.
10. For links in the Wizard Steps category, you can enter a text prompt for the link by opening the Prompt tab and entering the text in the text box. The prompt appears only on Select a Navigation Path Web pages.
11. Click **OK** to finish configuring the link.
12. Repeat these steps to add links for each of the destination nodes that users can reach from the current navigation node.

CONFIGURING INTERMEDIATE NODES

The state information that is carried across the link from one navigation node to another does not always contain all of the information that the Web page at the destination node requires. This is usually not an issue when the link behavior includes the Display component, in which case the destination Web page typically displays whatever state information it has. However, if the link behavior includes the Create Report component, you must ensure that destination Web page has the information that it needs. You can do so by creating a default report for each navigation node (which E.piphany recommends in any case). However, this method does not guarantee that the information the Web page derives from the default report meets the needs of different users.

To ensure that users get the results they desire, you can configure an intermediate node with a Web page of the appropriate Add FOAM type for your destination. Users can use the intermediate page to enter the information that they want to pass through to the destination page. When you create the navigation node for an intermediate page, use a link that has the Carry State, Display behavior from each originating node to the node for your intermediate page. Use a "Carry State, Create Report" link from the intermediate node to the destination node.

When you use the navigation nodes and links that have been supplied as part of a topic master, all of these transitions are configured correctly. Whenever you add new nodes to a topic, be sure to provide the appropriate intermediate Web pages and links for the transition from each origin to the node that you add.

The List Manager and Campaign Manager Web pages require that you provide intermediate pages whenever you wish to carry state from any of the reporting and analysis pages (other than Scoring), even across links with the Display behavior component. Other pages require intermediate pages only when a link specifies the Create Report behavior component.

Table 16 lists the destination Web pages for which intermediate nodes might be required. This table also shows the type of intermediate Web page that is required from each Web page type.

TABLE 16: INTERMEDIATE WEB-PAGE TYPES (1 OF 2)

	Destination Web Page Type	Intermediate Web Page Type	Origin Web Page Types (by row number in this table)
1	Advanced Rows and Columns	Add FOAM to Rows and Columns	3, 6,7, 9, 13, 15
	Basic Rows and Columns		
2	Campaign Calendar	n/a	n/a
3	Community Clusters	Add FOAM to Clusters	6, 9, 15
4	Cumulative Projections	Add FOAM to Rows and Columns	3, 6, 7, 9, 13, 15 (strongly recommended for 1, 11, 16)
5	Group Campaigns	Select Navigation Path to List Manager, Select Navigation Path to List Manager Popup	1, 3, 4, 7, 11, 13, 16
		Top N Scores	15

TABLE 16: INTERMEDIATE WEB-PAGE TYPES (2 OF 2)

	Destination Web Page Type	Intermediate Web Page Type	Origin Web Page Types (by row number in this table)
6	Group List Manager	Select Navigation Path to List Manager, Select Navigation Path to List Manager Popup	1, 3, 4, 7, 11, 13, 16
7	High/Low Clusters	Add FOAM to High/Low Clusters	6, 9, 15
8	Individual Campaigns	Select Navigation Path to List Manager, Select Navigation Path to List Manager Popup	1, 3, 4, 7, 11, 13, 16
		Top N Scores	15
9	Individual List Manager	Select Navigation Path to List Manager, Select Navigation Path to List Manager Popup	1, 3, 4, 7, 11, 13, 16
10	Influences	Add FOAM to Influences	6, 9, 15
11	Lifecycles	Add FOAM to Rows and Columns	3, 6, 7, 9, 17, 15 (strongly recommended for 1, 4, 16)
12	Modeling	Add FOAM to Influences	6, 9, 15
13	Profiling	Add FOAM to Profiles	6, 9, 15
14	Schedule	n/a	n/a
15	Scoring	n/a	n/a
16	Trends	Add FOAM to Trends	3, 6, 7, 9, 13, 15 (strongly recommended for 1, 4, 11)
17	View List	Add Transaction Type	1
	View List from Segment		

CONFIGURING NAVIGATION NODES FOR SCORING WEB PAGES

Follow these steps to configure navigation nodes that have the Scoring Web page type:

1. Create a Web Page of with the Scoring Web page type.
2. Select that Web page as the Web Page Contents for all navigation nodes with the Scoring Web page type.
3. Create at least two links in each Scoring navigation node:
   - one link in the New Model link category, to a node with a Modeling Web page, with the label "New Model."
   - the other in the New List link category, to a node with a List Manager Web page, with the label "New List."
4. E.piphany suggests adding additional links to additional nodes with other Web page types (such as Profiling) in the Analyze Further link category. These additional links allow users to perform additional analyses of data that they use for ranking list members.

CONFIGURING NODES FOR INFLUENCES, COMMUNITY CLUSTERS, AND MODELING WEB PAGES

You must ensure that the primary dimension used in any Influences, Community Clustering, or Modeling Web page that analyzes a list of individuals is the **indiv** dimension role. You must ensure that the primary dimension for any of these Web pages that analyzes a list of groups is the **group** dimension role.

CREATING INTERTOPIC LINKS

An intertopic link allows direct access from a navigation node in one topic to a navigation node in a different topic. Intertopic links can only be added in topics, not in topic masters. Aside from this, the process of adding intertopic links is the same as that for creating new links within a topic:

1. Edit the topic that contains the navigation node that is to be the origin of the intertopic link.
2. Click the **Add Link** button to display the Navigation Link dialog box. Select the destination node from the **Destination Node** drop-down list. This list box includes all of the navigation nodes in all of the topics that have been configured so far.

Note: *When you save a topic to a topic master, all intertopic links are excluded from that topic master. When you create a new topic from that topic master, you must create new intertopic links.*

PROVIDING ACCESS TO USERS AND GROUPS

You can give access to users and groups for specific nodes within a topic. This capability allows you to provide different views of the same topic to different users. If you do not give explicit access to a navigation node to a user or group, that person or group cannot view the Web page at that node.

Links from other navigation nodes to the node are also hidden from users and groups to whom you have not given access to a node. Links that originate at that node are also suppressed. This means that if a user or group does not have access to a node, and that node provides the only link to other nodes, the user or group is locked out of those nodes as well.

To provide access to an individual node:

1. Double-click the entry for that node in the Navigation Nodes tab of the Topic dialog box.
2. In the Access tab of the Navigation Node dialog box, click **Add Group, then** select the groups to whom you want to give access.

3. To deny access, select groups from the **Groups** list box and click the **Remove** button for groups.
4. Select the users to whom you want to give access, then click **Add User**. To deny access, select groups from the **Users** list and click the **Remove** button.

To provide access to a number of Navigation nodes to one or more groups:

1. Select all of the applicable navigation nodes in the Navigation Nodes tab of the Topic dialog box.
2. Click **Add User** or **Add Group**.
3. Select from among the available list of users and groups in the Choose dialog box, then click **OK**.

Adding Navigation Nodes

To add a navigation node to a topic, open the Navigation Nodes tab of the Topic dialog box for your topic, click the Add button, and then follow the instructions in the previous two sections for assigning Web pages and adding links.

Hiding or Deleting Navigation Nodes

When you have completed the process of assigning Web pages and links to the nodes that your application currently needs, you have a choice about how to handle nodes that the topic master has included but that are not needed in the particular topic you are creating. One option is to delete each unused node by selecting it in the Navigation Nodes tab of the Topic dialog box and clicking **Remove**.

Another option is to hide unused nodes by removing access to them for users and groups. For ease of maintenance, E.piphany suggests hiding nodes rather than removing them.

To remove access to unused nodes:

1. Double-click the row for the node you want to hide in the Navigation Nodes tab of the Topic dialog box, or select the row and click **Edit**.
2. Open the Access tab in the **Navigation Node** dialog box.
3. Choose all groups from the **Group Name** list and click the **Remove** button for groups.
4. Choose all users from the **User Name** list box and click the **Remove** button for users.

Whether you hide or remove an unused node, links from that node to destination nodes are disabled. If the route to a node that you need passes through a node that you remove or hide, that node is rendered inaccessible to users. You must ensure that there is an alternate link to every node that is the destination of a link from the current node. This is, of course, much easier to do before you delete the node. If you hide the node instead, you can still view the destinations of its links in EpiCenter Manager, even though those links have been rendered inactive for end users.

CONFIGURING LIST-MANAGER NAVIGATION NODES

When you configure a node with a Group List Manager or Individual List Manager Web page, you must configure a link with a List Preview Web-page type. The Web page that you choose for the Web-page content at the destination link can be any Web page of the same Web-page type.

PROVIDING ON-LINE HELP FOR TOPICS AND WEB PAGES

Each E.piphany solution includes a set of on-line help pages. These Web-based pages describe the functionality of the various Web pages and give a generic overview of E.piphany solutions. E.piphany also provides a number of Web-based help pages that you must customize to provide site-specific information about the Web pages and topics that you configure, and the solution maps that you install:

- **Topics.html** and **Topic1.html**

  You can customize the **Topics.html** file to display a list of the topics that you have configured, with links to help pages for each individual topic. The **Topic1.html** file is a template that you can copy and customize to create individual help pages for each of the topics that you have configured.

- **SN_*.html**

  You can update these files to display local annotations to each of the standard help files that are part of your E.piphany application.

- **Solutions.html**

  You can update this file to display a list of the E.piphany solution maps that have been installed as part of your application, with links to E.piphany-supplied help pages that describe these solution maps. The help pages for individual solution maps are installed along with the solution-specific software that you install.

These files reside in the following directory:

```
C:\Program Files\Epiphany\Instance\WWWRoot\Help\SiteNotes
```

Replace *C* with the drive on which your E.piphany software is installed and *Instance* with the name of your E.piphany instance. These files are in HTML format. E.piphany suggests that you use a text editor to copy, update, check, and then replace the raw HTML files, in order to avoid potential formatting problems that can be caused by some HTML authoring packages.

These help pages are available to all users of your E.piphany application. If you want to restrict access to information about a specific topic, you can use a node with a Navigation Path Web page as the beginning node of that topic, and enter the descriptive text as a prompt for that node.

CREATING AND EDITING TOPIC MASTERS

When you have completed a topic that has the right combination of navigation nodes, Web pages, and links, you can save it as a topic master by clicking the **Save As Master** button in the Topic dialog box. When you create a topic master from an existing topic, the new topic master includes the navigation nodes, Web page types, and links. The specific Web page contents are not included in the topic master.

To edit a topic master:

1. Open the **Solutions** folder in EpiCenter Manager, then the **Topic Master** folder, and then double-click the topic master you want to edit to display the Topic Master dialog box. The General tab of this dialog box displays the name of the topic master and the navigation nodes that the topic master includes.

Note: Topic masters for E.piphany solution maps are not editable.

2. Add, edit, or delete nodes as appropriate:
   - Use the **Add** button to add a node to the topic master. This button displays the Navigation Node dialog box (Figure 109, on page 339). Follow the instructions in "Configuring Navigation Nodes," on page 337 to enter the appropriate information about the node itself, and the links you wish to assign to the node.

Note: The Access and Reports tabs do not appear in the Navigation Nodes dialog box that is displayed for a topic master.

- Choose a node from the list and then click the **Edit** button (or double-click the node entry) to display the Navigation Node dialog box, which you can use to update information about the node and the links that have been assigned to it.
- Chose a node from the list and then click the **Remove** button to delete that node and all of the links that originate at that node.

3. Choose a beginning node by selecting a node and checking the **Beginning** check box. Then choose a behavior for the link from the Home page to the beginning node. E.piphany suggests that you choose the **Carry State, Display** behavior for this link.
4. Enter a description for the topic master in the Description tab.

The Usage tab of the Topic Master dialog box displays the list of topics that have been derived from a topic master. You can update the topics that have been derived from the current topic master to reflect the edits that you have made in the topic master by selecting the each topic in EpiCenter Manager and clicking the **Update** button in the General tab of the Topic dialog box.

REFRESHING A TOPIC FROM AN UPDATED TOPIC MASTER

When you update a topic master, EpiCenter Manager does not propagate the changes that you have made in the topic master to existing topics that have been derived from that topic master. To refresh the structure of a topic from its updated topic master, double-click that topic in EpiCenter Manager, and then click the **Update** button in the Topic dialog box for that topic.

If the extent of the updates that you have made to a topic master reaches a level at which you would prefer to rebuild an existing topic rather than update it, you can delete the topic and then create it once again. However, whenever you delete a topic, EpiCenter Manager deletes all of the saved reports that are assigned to nodes within that topic.

Warning: Deleting a topic deletes the saved reports that are assigned to nodes within that topic.

You can preserve your save reports by exporting them, rebuilding your topic, and then importing them again, as follows:

1. Right-click the topic in EpiCenter Manager, then click **Export Saved Reports**.
2. Enter the name of a file in which to store the exported metadata for your saved reports in the Export dialog box, and then click **Go**.
3. Delete the topic.
4. Create a new topic that has the same name as the topic that you just deleted. If you want to rename the topic, do so after you have completed these steps.
5. Import saved reports from the file of exported metadata.

THE E.PIPHANY APPLICATION SERVER

The E.piphany Application Server (AppServer) is the middleware component that allows users to interact with an EpiCenter datamart through the E.piphany Web-based user interface. The AppServer accepts requests that users submit with E.piphany Web pages, dispatches queries to the datamart, and returns the results of those queries to users in the form of Web pages that contain query results or responses.

Not all user requests require queries against the datamart. For instance, the following types of requests, among others, are processed by the AppServer itself:

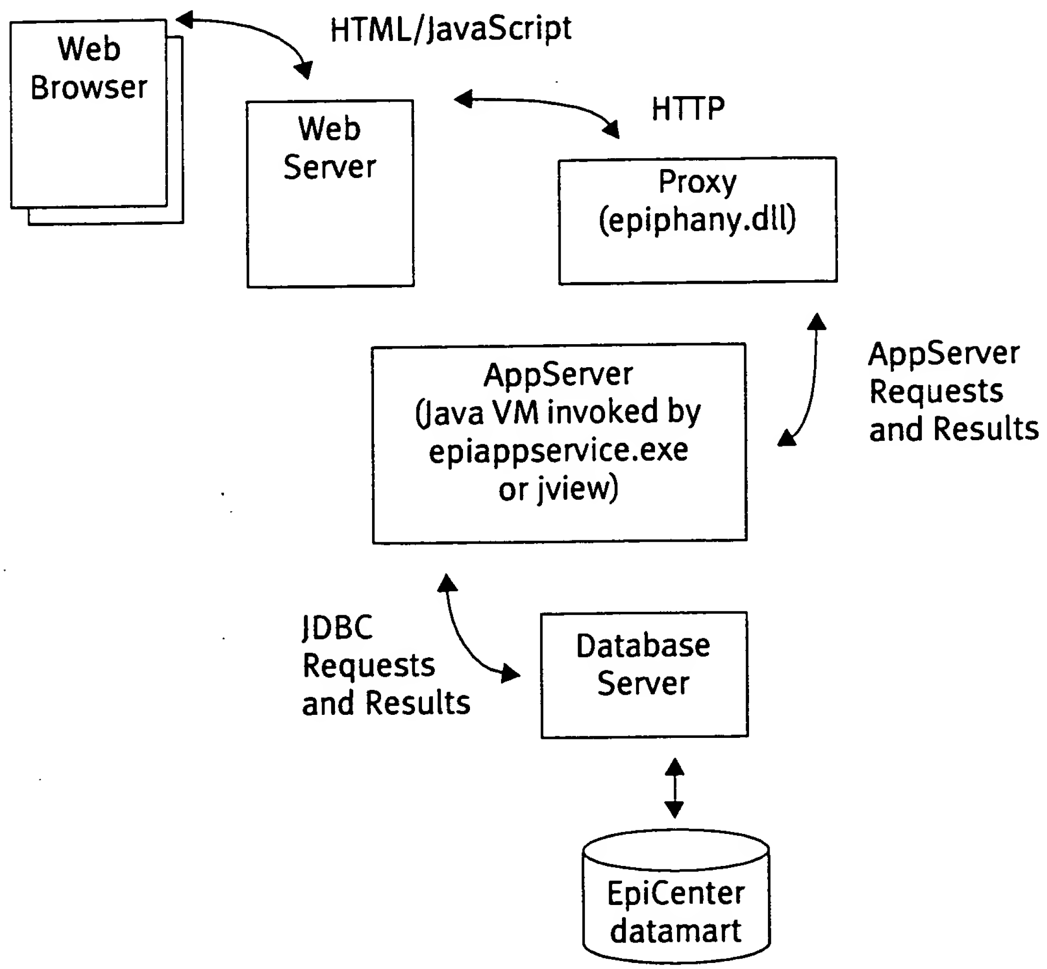
- Requests to log in to the E.piphany application
- Requests to display the Web page at a particular navigation node
- Requests to browse the Report Gallery

The AppServer is a Java application that manages user connections and database queries in a multithreaded fashion. Figure 112, on page 358, illustrates the role that the AppServer plays in the architecture of your E.piphany application.

The AppServer is implemented as a collection of Java classes that run within a Java virtual machine. The virtual machine that runs the AppServer code can be invoked as a Windows NT Server service, or directly as a **jview** command.

The AppServer connects to the Web server by means of a proxy program called **epiphany.dll**. The Web server passed HTTP requests to this proxy, which forwards those requests to the Java virtual machine that executes the AppServer code. The proxy isolates the AppServer from the Web server for greater reliability of your E.piphany application.

FIGURE 112: ROLE OF THE APPLICATION SERVER


Web Browser
HTML/JavaScript
Web Server
HTTP
Proxy (epiphany.dll)
AppServer (Java VM invoked by epiappservice.exe or jview)
AppServer Requests and Results
JDBC Requests and Results
Database Server
EpiCenter datamart

An E.piphany user session begins when a user requests the URL for an E.piphany proxy as the destination to visit in a Web browser. The URL request is resolved by the Web server (IIS), and then routed to the E.piphany proxy, **Epiphany.dll**. The proxy is an Internet Server Application Programming Interface (ISAPI) application that packages the request and forwards it to the Application Server through a TCP/IP port.

The proxy handles requests by other users as it waits for the AppServer to process the request and return the result, which is the data to display the E.piphany login page. The proxy then forwards this result to the Web server that, in turn, routes that result to the user's Web browser for display. For more information about this proxy, refer to "The E.piphany Proxy," on page 380.

An E.piphany installation can include multiple Application Servers that each communicate over different TCP/IP ports. For example, a second Application Server might serve as a backup, or one Application Server might be used for development and another used for production.

For instructions on how to troubleshoot the Application Server, see Appendix H, "Troubleshooting." That appendix also describes Application-Server error messages.

STARTING AND STOPPING THE APPLICATION SERVER

The standard E.piphany software installation (as described in the *E.piphany e.4 Installation Guide*) configures the E.piphany Application Server as an NT service. You can start and stop the service using the Windows **Control Panel\Services** menu by following these steps:

1. From the Start menu, choose **Settings\Control Panel\Services**.
2. Select the E.piphany Application Server. The name of the service is the *instance_name* that you specified during installation.
3. Click Stop to stop the Application Server, or click Start to start it.

Tip: *The AppServer checks for inconsistencies in the metadata for your E.piphany application before it begins processing requests. Such inconsistencies can prevent the AppServer from starting successfully. The first step in diagnosing such a failure is to run the Scrutiny debugging tool. See "Running the Scrutiny Debugging Tool," on page 299, for details.*

MONITORING THE APPLICATION SERVER

To determine if the Application Server is running, enter the URL for the E.piphany proxy as the destination to visit in a Web browser. If the browser displays the E.piphany login page, the AppServer is running. If the Web browser displays an E.piphany proxy error message, you can attempt to start the AppServer by following the instructions in the previous section.

If the Web browser appears to stall or hang up, you can check the following resources to see if it is still running:

```
http://www.hostname.com?monitor
```

Replace *hostname* with the name of the computer on which the AppServer is supposed to be running.

1. Use the Web-based monitoring interface to query the AppServer. Enter a URL of the following form as the destination for your Web browser, as discussed in the next section
2. Check if the service is running in the Control Panel.
3. Check the NT Event log by going to **Start\Programs\Administrative Tools (common)\Event Viewer** and opening the **Log\Application** menu.

   If the Service was started and is running, there will be an entry with `Source = EpiAppServer`. The message reads:

   ```
   Epiphany Appserver instance message:
   The Service was started.
   ```

   Check for the existence of the AppServer log file. By default, this file resides in:

   ```
   C:\Program Files\Epiphany\instance\web\WWWRoot\logfiles
   ```

   or in the directory specified by the following registry key:

   ```
   HKEY_LOCAL_MACHINE\Software\Epiphany\Instances\
   instance\SystemLogDir
   ```

The log file has a name in the following date-time format. The filename indicates the year, month, day, hour, minute and second when the Application Server started running.

```
YYYY-MM-DD_hh-mm-ss-32SRV.txt
```

USING THE WEB-BASED APPSERVER MONITOR

E.piphany provides you with a Web-based interface for monitoring and refreshing the AppServer. This interface displays current status and activities, and allows you to browse the various logs that the AppServer creates. To log in to the AppServer monitor, supply the following URL as the destination for your Web browser:

```
http://host/script/instance/epiphany.dll?monitor
```

Replace *host* with the name of the host machine on which the AppServer runs. Replace *instance* with the instance name by which your E.piphany application has been installed.

Note: *You must be a member of the Administrator group for your E.piphany application in order to log in to the AppServer monitor.*

The AppServer monitor Web page is divided into the following topics:

- General

  This section provides summary information about system resources that are currently being used by your E.piphany application.

- Current Activity

  This section provides a link that you can use to view the current activity log.

- Statistics

  This section provides a link that you can use to display statistics.

- Sessions

  This section provides a link that you can use to view the session logs for individual users.

- Refresh

  This section provides a link that you can use to refresh the AppServer.
- Debugging

  This section provides a link that you can use to track threads.

The AppServer monitor allows you to examine current sessions and associated states. Click on a session to see a list of states that are currently loaded by the AppServer for that session. Click on a state to see the **Params** structure, a list of key-value pairs that describe the state.

REFRESHING THE APPLICATION SERVER

Upon startup, the Application Server reads metadata and Registry information and caches it for use during normal query processing. The metadata that is read includes:

- Information about the navigation nodes, link behaviors, Web pages, and Web-page templates
- Datamart configuration information
- The Windows Registry
- Aggregate navigation information.
- Time navigation information.
- Security information
- Storage information
- Scheduling information

When any of the following events occur, you need to restart or refresh the Application Server:

- After an extraction, either the **current_datamart** value has changed or there has been a change in a table that is used in a dynamic listbox filter
- If your EpiCenter datamart resides in an SQL Server database and the *MSSQL Server* service stops, you must restart both that service and the AppServer.
- EpiCenter Manager has been used to reconfigure your E.piphany application
- Aggregates have been built or rebuilt since the last time the Application Server was started
- The Windows Registry entries under the following entry have changed since the last time the Application Server was started:

  ```
  HKEY_LOCAL_MACHINE/Software/Epiphany/instance_name
  ```

If you have determined that you need to refresh the Application Server, there are several ways to proceed. You can stop and then restart the Application Server via the Services Control Panel. However, this requires manual intervention, and is therefore unsuitable for programmatic refresh (after an extraction, for example). This method also interrupts anyone currently using the system.

Other ways to refresh the AppServer that do not interrupt users include:

- Using the **refresh.exe** command line
- Using the Web-based AppServer monitor. The URL for the AppServer monitor for an E.piphany application takes the following form:

  ```
  http://hostname/scripts/instance/Epiphany.dll?monitor.
  ```

  Replace *hostname* with the name of the computer on which your E.piphany application resides. Replace *instance* with the name of your E.piphany application instance. To display the AppServer monitor, enter the URL in the **Address** or **Location** text box of your Web browser.

The AppServer monitor allows you to refresh the Application Server. It also allows you to monitor the current state and recent activity of the AppServer. To refresh the AppServer, click the **Refresh the Application Server** link. If the operation successful, the Web page displays the following message. Otherwise, the Web page displays an error message.

```
AppServer was refreshed
```

Refreshing does not interrupt the requests that are currently running on the Application Server.

Another way to refresh the server is by using the **refresh.exe** program that was installed in the **win32** subdirectory of your E.piphany installation directory. This program sends a message to the Application Server instructing it to reread all of the necessary information mentioned above.

You can use the **refresh** command to refresh the AppServer.

REFRESH

refresh ***localhost port username password***

This command refreshes an E.piphany AppServer.

localhost	specifies the name of the machine on which the Application Server is running.
port	specifies the port number on which the Application Server is listening.
username ***password***	The *username* parameter specifies a valid administrative account. The *password* parameter specifies a valid password for that account. These are the same values required for an end user to log into the top-level E.piphany Web page. In general, any account that is defined in the Security folder of the EpiCenter Manager application will work. See “Security,” on page 273 for more information about E.piphany users.

After the **refresh** command has established a connection to the AppServer, it displays the following message and waits for a response:

```
Sent the REFRESH instruction to localhost
```

Normally, the Application Server takes about 30 seconds to refresh before returning an acknowledgment. (Times may vary based on the size of your EpiMeta database, the speed of your network, and other considerations.) Upon receiving this acknowledgment, the **refresh** program displays:

```
Refresh [Build 4.0.x.y]
---------------------------------------
Connecting to localhost:8081
Sent the REFRESH instruction to localhost
The refresh SUCCEEDED.
REFRESH operation took 18827 ms.
```

Otherwise, **refresh** displays a failure indication. For example, here is the output of a negative interaction in which the user/password failed.

```
Refresh [Build 4.0.x.y]
---------------------------------------
Connecting to localhost:8081
Sent the REFRESH instruction to localhost
The refresh FAILED because the username/password combination was
invalid.
REFRESH operation took 3365 ms.
```

The **refresh** program always displays the amount of time that it has taken.

MANAGING APPSERVER SERVICES

You can use the EpiAppService program **(epiappservice.exe)** to manage one or more AppServer services that run on the local host. This command allows you to perform the following actions:

- Add a new AppServer service
- Start or stop an AppServer service
- Change the parameters of an AppServer service
- Delete an AppServer service

The syntax of the **epiappservice** command follows.

EpiAppService

(1 of 2)

epiappservice -s -C -D -G -T -P -K [-DEP *svc*] [-E *cmd*] *servicename action*

This command registers an E.piphany AppServer as a service under Windows NT Server.

-s specifies the name of the machine on which the Application Server is running.

-C Creates a service. Registers a new service with the NT system. You must use the **-E** option with this option to specify the executable to use as the NT Service; for example:

```
EpiAppService -S "instance" -C -E "program name"
```

When you are configuring an E.piphany Application Server, the command line looks like this:

```
EpiAppService -S "instance" -C -E "C:\Program
Files\Epiphany\instance\win32\EpiAppService.exe -S
instance"
```

If the instance name has spaces in it, then you should use single quotes inside of the outside quotes. Do not use spaces within the instance name.

-D Deletes the service specified via the **-S** option from the NT Service Registry. When this service is deleted, only the entry in the NT Registry for the service is deleted; no executables are deleted. Use this to remove unused Application Server instances.

-G The launch handler for the service. (Go.)

-T sTarts a service. The installation program uses this option to start the Web server (which must be shut down during installation). It starts the service specified via the **-S** option, which is equivalent to clicking the Start button in the **Control Panel\Services** dialog box to start the desired service. You may use this option when you cannot access the Control Panels (for example, during **RCMD** or **pcAnywhere** remote access).

-P stoPs the service specified via the **-S** option. (See the **-T** option.) The installation program uses this option to stop the Web server during the installation.

EPIAPPSERVICE (2 OF 2)

-K	Checks if the service specified via the **-S** option is running. The program returns a zero return code if the service is running, and a nonzero return code if it is not.
-DEP ***svc***	The *svc* argument specifies a service on which the Application Server depends. The Service Manager will always load this dependency before loading the Epiphany Application Server. The service name supplied as an argument to this option must exactly match the service name used by the service in question. such as Microsoft SQL Server or the Oracle listener.
-E ***cmd***	The *cmd* argument specifies a command to associate with the service.
-?	Displays online help, a description of the command options.

The following examples illustrate some common uses of the **eppiappservice.exe** command:

- Creation of a service because the installation failed, or you wish to install another instance. If the database server is on a different machine from the Application Server:

  ```
  EpiAppService -S instance -C -E "C:\Program Files\
  Epiphany\instname\Win32\EpiAppService -S instname"
  ```

  If the database server is on the same machine as the Application Server:

  ```
  EpiAppService -S instance -C -E "C:\Program Files\
  Epiphany\instance\Win32\EpiAppService -S instname" -DEP
  MSSQLServer
  ```

- Deletion of an old service.

  ```
  EpiAppService -S instance -D
  ```

RUNNING THE APPSERVER AS A CONSOLE APPLICATION

You can use the **jview** command to start the Application Server from a command console if you have the following Windows NT Server privileges:

- Act as part of OS
- Increase quotas
- Replace a process level token

Note: *You can use the pass through security module without setting these privileges. See "Authentication Modules," on page 374.*

The Local System account already has these privileges, and E.piphany recommends that you configure your Application Server to run as a service under the Local System account.

To run the Application Server from the command line, follow these steps:

Note: *This procedure affects only the local machine, not the NT network.*

1. Log in as a user who has local administration access.
2. Assign special NT privileges to a user account. To do so, start User Manager and enter the local machine name in the Select Domain dialog box.
3. Choose **Policies\User Rights** from the menu.
4. Click Show Advanced User Rights and assign the account of the currently logged-in user to have the privileges specified above.
5. Reboot your machine for the privileges to take effect.
6. In a command window, navigate to the following directory:

```
C:\Program Files\Epiphany\instance\classes
```

7. Enter **server** to run the **server.bat** file, or enter the following command to start the AppServer:

```
jview /cp:p C:\Program Files\Epiphany\instance\EpiAppServer.jar
com.epiphany.server.Server @instance
```

If the machine name and port number are not specified, then they are read from the Registry under the registry key for the instance that you have specified. If the *instance_name* is not specified, then the AppServer reads the default value in the following registry key to determine the instance name to use:

```
HKEY_LOCAL_MACHINE\Software\Epiphany\Instances
```

When you start the AppServer from the command line, log information is copied to a console window. This console window must remain open while the Application Server runs. If you close the console window, the Application Server terminates. If you log off, the console window closes, and the Application Server terminates.

APPLICATION SERVER LOGGING

Most components of the Application Server maintain a log file of their activities. This section describes each of these logs, their directory location, and their diagnostic use. In the Windows NT Registry on the Application Server machine, the Registry key that specifies the location of all log files generated by the Application Server is named:

```
HKEY_LOCAL_MACHINE\Software\Epiphany\Instances\instance_name
\SystemLogDir
```

All log file names begin with a prefix that indicates the date and time when the log file was first created. The format is as follows:

```
YYYY-MM-DD_HH-MM-SS-MStype.txt
```

YYYY stands for the year, *MM* for the month, *DD* for the day, *HH-MM-SS-MS* for the time (hour, minute, second, and millisecond), and *type* stands for log file type, as listed in Table 17, on page 370. The logs of queries for E.piphany applications have the suffix `QM_sessionid`. *Sessionid* stands for the session ID of the particular Web page from which the query originated.

TABLE 17: APPSERVER LOG TYPES

Log Type	Description
Server Log	Logs all connections made to the Application Server. This file contains the time at which a connection was accepted. the numher of that connection, the time at which the request was finished. the total time required by the request, and certain messages printed to the log during the processing of that request. The latter category includes: the session ID of the connection, the template used to process the request, the number of bytes generated in the response, exceptions that might be generated, and the user name.
Connection Log	Logs all function calls, SQL, exceptions, and errors related to database-server connections.
Schedule Log	Logs all function calls, SQL, exceptions, and errors for the schedule subsystem.
SecurityManager Log	Logs all function calls, SQL, exceptions, and errors for the security subsystem.
StorageManager Log	Logs all function calls, SQL, exceptions, and errors for the storage subsystem.
Action Log	Each Web-page request generates a log that contains Web-page parameters, all of the SQL statements that have been executed during the processing of the request, and information about how long the processing took.

CACHING OF QUERY RESULTS

The AppServer caches results of queries that are run against the datamart. If another query uses the same dimensions, measures, and filters, the AppServer uses the data it has cached rather than issuing another query against the datamart. An optional registry key, **QueryCacheSize**, determines the number of query results that the AppServer caches. If this key is not set to another value, the AppServer caches the most recent 200 query results.

TRACKING OF QUERIES FOR AGGREGATE OPTIMIZATION

The AppServer maintains a log in metadata for the tables and columns that queries access. This log is used by the aggregate optimizer to generate statistics and make recommendations about which aggregates to build. You must run one of the following SQL scripts to clear these logs from time to time, in order to avoid filling the device or tablespace in which your EpiMeta database resides.

```
querylogtrunc_mssql.sql # for SQL Server

querylogtrunc_ora.sql # for Oracle
```

These SQL scripts are located in the following directory:

```
C:\Program Files\Epiphany\instance\Configfiles
```

When you run this script, the data that the aggregate optimizer uses to make recommendations for new aggregates is removed. The next time you refresh the aggregate optimizer from the logs, all historical data prior to the time at which the script has been run is lost. E.piphany recommends that you refrain from refreshing the aggregate optimizer for an appropriate number of days after running this script to reestablish a history of recent queries.

APPLICATION SERVER SECURITY

Authentication, which is performed outside of the E.piphany system, does not capture password information. This external authentication requires an authentication module. For example, the NT authentication module authenticates users with an NT domain controller. The Security Manager inside the Application Server loads the authentication module specified through the SecurityClass Registry key at initialization. Each authentication module supports a fixed API that includes methods to authenticate users, add new users to the E.piphany system, and sync-up outside information on the user (such as group memberships) with E.piphany metadata.

Currently, E.piphany provides two authentication modules: the NT authentication module EpiNTLogon, and an insecure authentication module called EpiPassThruLogon. Use EpiPassThruLogon only for testing and debugging. Optionally, you can add an LDAP authentication module, X.500 module, and other similar modules.

The optional Registry key SecurityClass under the *instance_name* Registry directory controls which security module is loaded. You must specify the full class path to the security module. If this key is omitted, the system uses the default security module, **com.epiphany.security.EpiNTLogon**, which uses NT to perform user authentication.

The means by which the authentication information (username and password) reaches the Application Server is as follows. Users log into the E.piphany system either through a login template or through a Web server authentication mechanism, such as Basic Authentication or NTLM.

When the Application Server receives the user name and password, it calls the Security Manager to log in the user. The Security Manager loads an authentication module, and attempts the login process. If the process is successful, depending on the authentication module, one of the following steps is taken:

- If the user exists in the EpiMeta database, user group memberships outside of the E.piphany system will be synched up with group memberships in the EpiMeta database.
- If the user does *not* exist in the EpiMeta database, but is authorized to use the E.piphany system, then a user account will be created in the EpiMeta database. User group memberships outside of the E.piphany system, such as NT, will be synced-up with group memberships in the EpiMeta database. For example, assume your EpiMeta had a group named EPIPHANYUSERS configured with access to all of the Web Pages. If a user name Joe (a valid NT user who belongs to the NT group EPIPHANYUSERS) supplies a correct password, then Joe will be automatically added to the EpiMeta metadata as a user who belongs to the E.piphany group EPIPHANYUSERS.

Only group memberships for a group whose Group Definitions dialog box in EpiCenter Manager has the Synchronize option selected will be synced-up. Syncronization occurs when:

- The user who logs in is a member of a Group X outside of the E.piphany system, and:

  a) that same group is defined within the E.piphany system

  b) the user is not a member of that group inside the E.piphany system.

  In this case, sync-up adds a group membership to the E.piphany system.

- The user who logs in is a member of a Group X inside the E.piphany system, but the user is *not* a member of this group outside of the E.piphany system.

  In this case, sync-up removes a group membership from the E.piphany system.

If the sync-up process requires a creation of a new group membership for a user, certain access rights are set up based on this membership. The ability to save queries has the access rights of Save Group/Default, and global-level save access is Inherit. (See "Security," on page 273 for more information.)

For a group to be the same in EpiMeta and the NT domain, it needs to have the identical name (case sensitive) in both. The group name in the NT domain includes the domain name, such as **Epiphany\Sarah**. You need to make sure that the group name in the EpiMeta includes the domain name for that group.

When the user is authenticated and user information is synced-up, Security Manager determines if the user is authorized to use the E.piphany system. The user is authorized when the password is correct and he or she belongs to at least one group in the E.piphany system.

AUTHENTICATION MODULES

The following authentication modules are included with the E.piphany software.

Module	Class Name
NT (default)	**com.epiphany.security.EpiNTLogon**
Pass through	**com.epiphany.security.EpiPassThruLogon**

- NT (default)

  NT domain authentication. NT groups are imported into the E.piphany system through EpiCenter Manager. As mentioned, these groups need to have the Synchronize option selected in the Group Definition dialog box, which means that memberships to those groups will be synced-up. Users who belong to those groups can log into the E.piphany system. When a user logs in, his or her NT group memberships are synced-up to the EpiMeta database. It is also possible to create E.piphany-only groups inside EpiCenter Manager by omitting the Synchronize option. Thus, the E.piphany administrator can create arbitrarily complex permission hierarchies using EpiCenter Manager, independent of the manner in which NT domain security is set up.

- Pass through (*for in-house debugging purposes only; should never be used at a customer site after the initial E.piphany system setup*)

  This is an insecure authentication that has no password. This is an E.piphany-only development module that ignores NT authentication altogether. You do not need passwords in this model: specify your user name exactly as it appears in EpiCenter Manager (it must include a domain name if such exists), and you will be logged in. There is no sync-up process. That means that an authenticated user that does not exist in the EpiMeta database will not be allowed to use the E.piphany system. E.piphany groups and users are created and managed through EpiCenter Manager.

AUTHENTICATION MODULE TIPS

- Users that do not belong to any groups in the E.piphany system are not allowed to log in. An error message that says that user is authenticated but not authorized to use E.piphany system is displayed whenever an unauthorized but NT-authenticated user logs into the Application Server. User memberships may be adjusted upon login if the user is a member of synchronized groups in E.piphany system. A user must be a member of at least one E.piphany group after the synchronization process completes.
- The optional Registry key SecurityClass (located in the *instance_name* Registry directory) controls which security authentication module is loaded. The full class path to the security module must be specified as the value for this key. If this key is omitted, the default security module is **com.epiphany.security.EpiNTLogon**, which uses NT to perform user authentication.
- **Toplevel.template** is the template that appears immediately after a user logs into the E.piphany system. The name of this template is configurable with the an optional ToplevelTemplate Registry entry in the **Instances** directory. You can load **company.template** instead of **toplevel.template** by substituting your company name in `ToplevelTemplate=`*company*.
- Depending on how IIS security is set up, one of the following situations occurs upon login:
  - If Allow Anonymous authentication is enabled, the login dialog box is displayed when the user attempts to use the system for the first time, or the user session times out. It is almost always sufficient to specify the username only. The domain name is located automatically.
  - The search order that authentication uses to find the user account is as follows. First, the authentication mechanism looks in the local machine's Security Access Manager (SAM), then it checks with the primary domain controller, and afterwards checks with trusted domains. If there are multiple users with the same name between domains and/or a local machine that runs the Application Server, specify the full user *name-domainname\username*, or the *local_machine_name\username* if the user logs in from a local machine's Security Access Manager (SAM).

- If Basic Authentication is enabled, the browser displays a login dialog box when the user attempts to access the E.piphany system for the first time. However, when the user's session times out, no re-login is required. The user is automatically logged in again, and a new user session created.
- If NTLM authentication is enabled, Internet Explorer automatically performs authentication of the user without displaying a login dialog box, although Netscape Navigator displays it. If Basic Authentication or NTLM is on, the login dialog box does not appear in the Web browser.

Warning: Do not create your own domains. Doing so introduces multiple domains, with the E.piphany machine in one domain and the user accounts of the E.piphany system in another domain. In order for the authentication module ***com.epiphany.security.EpiNTLogon*** *to work properly, a two-way domain trust between the E.piphany domain and the customer domain is required.*

If you set up a new domain for the machine that runs the E.piphany Application Server, set up a two-way trust and name the machine and the domain differently. In general, do not use the same string for domain names, machine names, and user names.

- The following information applies to Synchronized groups:

  EpiNTLogon requires the NT domain for lists of global and local groups. The names of these groups will be matched to the names of the groups in the EpiMeta. A group name will be matched if its domain name and group name match. Therefore, group *xyz* will not match to group *EPIPHANY\xyz*. The match is case insensitive for both group name and domain name. If a user is a member of an NT group *EPIPHANY\xyz* and EpiMeta has a group called *epiphany\XYZ*, a match occurs.

  - If a user is a member of a local group and the group has a global group as a member, the global group will not be picked for the synchronization process. Only the groups for which the user is an immediate member are considered for synchronization.

- If a user logs in with an account that is local to the Application Server machine, then a membership to a special group called None is automatically retrieved from the machine's SAM. (Every user in the local SAM has a membership in a special global group called None although this group does not exist on the machine.) For this reason, do not create synchronized groups called None in EpiCenter Manager.
- Avoid having the same name for domain names, machine names, and user names. For example, if the Application Server runs on the machine *xyz*, and the user called *xyz* attempts to log in, access may be denied. If the Application Server is running on a machine named *xyz*, and a user named *xyz* logs in from the primary domain, or from the local Security Access Manager (SAM) of the machine *xyz*, authentication will succeed. If user *xyz* logs in from a trust domain, however, the authentication will fail. The only way to log in as *xyz* from another domain is to give the full name for the user account upon login: *domainname\xyz*.

ADMINISTRATOR GROUPS

Epicenter Manager users can make any group an administrator group, and there can be multiple administrator groups in the system. If the user belongs to such a group, that user has special powers when it comes to report/folder and Web-page access.

Administrator users can save, overwrite, create, and change properties and permissions on any nonspecial and nonhidden file or folder.

Note: *Special folders are top-level folders, such as Public and All Users, or user/group folders, or default folders. Hidden folders are the MailTo folder. Hidden files are clipboard files.*

Administrator users have access to all Web Pages in the system.

APPLICATION SERVER REGISTRY KEYS

All Registry keys used by the E.piphany Application Server are located in:

HKEY_LOCAL_MACHINE/Software/Epiphany/Instances/*instance_name*

where *instance_name* specifies the name of the instance you entered during the E.piphany software installation. Table 18 lists the registry keys in this directory:

TABLE 18: APPSERVER REGISTRY KEYS (1 OF 3)

Registry Key	Description
AppServerHost	The name of the machine on which the Application Server is running The default host is the local host.
AppServerLogVerbosity	The verbosity level for database logging performed by the AppServer If the value is 0, no logging is performed. If the value is 1, logging occurs.
AppServerPort	The port number on which the Application Server is listening for connections The default port is 8081.
AppServerQueryTimeout	The default timeout value, in seconds, before the Application Server automatically terminates long-running queries A query that exceeds the indicated number of seconds is terminated in order to prevent undue resource contention. This value can be overridden by a user, for her or his queries only, in the User Preferences dialog of the Home page.
AppSessionTimeout	The value in seconds for the lifetime of an idle session A session that remains idle for longer than this time is removed from the cache.
AppStateTimeout	The threshold, in seconds, at which Web-page state information is swapped from AppServer memory to disk
ChartsOutputDir	The directory (full path) in which the ***.epc** chart files will be stored

TABLE 18: APPSERVER REGISTRY KEYS (2 OF 3)

Registry Key	Description
DatabaseName	The name of the EpiMeta database
DatabasePassword	The password that corresponds to the **DatabaseUsername** account
DatabasePort	The port that supports the connection to the database server The default port for SQL Server is 1433. The default port for Oracle is 1521.
DatabaseServer	The name of the database server on which the EpiMeta database is located
DatabaseType	The type of the database
DatabaseUsername	A username for logging into both the EpiMeta and EpiMart databases
Description	A textual description of the instance
InstanceRootDir	The root directory into which the E.piphany Application Server has been installed
MaxDBConnections	The maximum number of simultaneous open connections to the database server. The default number is 100
MomentumOutputDir	The directory in which campaign output files are stored
ProxyLogFile	(Optional) Enables proxy logging For example, setting this key to **C:\proxy.log** will create a **proxy.log** file if it does not already exist, and log information for every proxy submit to the Application Server. If an invalid path is specified in this ProxyLogFile Registry key, no logging will be performed.
QueryCacheSize	The number of most-recent query results to store in the AppServer cache
ScrutinyDisabled	(Optional) When present and set to `Yes`, disables the Scrutiny checks on metadata that are automatically performed when the AppServer starts up

TABLE 18: APPSERVER REGISTRY KEYS (3 OF 3)

Registry Key	Description
SecurityClass	(Optional) The Java class to use for security authentication The default class is **com.epiphany.security.EpiNTLogon**.
StateDir	The directory in which Web-page state information is temporarily stored
SystemLogDir	The directory in which the E.piphany Application Server log files are written
SystemLogDirWebpath	The name appended to **http**:*//machinename/instance_name/* that informs the Web server of the locations of log files.
TempDirGarbageLifetime	The lifetime in seconds of a temporary or log file created by the E.piphany Application Server After this specified number of seconds from the creation date, a temporary file, log file, or ***.epc** chart file will be erased by the Temporary File Manager in the Application Server. Garbage collection occurs at Application Server startup and periodically when there are free cycles.
TemplateDir	The directory in which the E.piphany Application Server templates are stored

WWWRootDir	(Optional) The path to the E.piphany instance Web directory, for example:
	`C:\Program Files\Epiphany\`
	`instance\Web\WWWroot`

THE E.PIPHANY PROXY

The E.piphany proxy (**Epiphany.dll**) is an ISAPI application that mediates the requests and responses between the IIS Web server and the E.piphany Application Server. All user requests for E.piphany pages are directed to the proxy:

```
http://machinename/scripts/instance_name/Epiphany.dll
```

This proxy bundles the request into a package that conforms to a strict E.piphany format and sends the package to the E.piphany Application server through a TCP/IP socket. The proxy uses the Windows Registry to find the E.piphany Application Server. In particular, the proxy parses the *instance_name* from the requesting URL. It then opens the AppServerHost and AppServerPort found in the Registry tree for that instance, and uses this information to connect to the Application Server. The Application Server processes the request and sends a result back to the proxy. The proxy displays the result in the user's browser.

Note: *The instance_name in the URL must match the instance_name as defined in the Windows Registry (configured from the installation program). This allows one proxy to direct requests to several different Application Server instances. This is of value when you want to have two instances, such as a release and a test instance.*

PROXY LOGGING

The **Epiphany.dll** proxy supports rudimentary logging. If you suspect that the proxy is not passing all parameters to the Application Server, you can enable logging of all parameters that the proxy submits.

Proxy logging, which is optional, is a diagnostic tool for identifying problems, not a run-time logging facility. Because the log file may grow arbitrarily large if proxy logging is always enabled, use it only in the event of a suspected problem with the **Epiphany.dll** proxy.

An example of a proxy log file:

```
Processing new request. Header: mGET q·p v0.8 oHTTP/1.0 nzhenya P80
r192.0.0.147 aGodzilla/4.04 [en] (WinNT; I ;Nav) u
U/scripts/capri/Epiphany.dll S/scripts/capri/Epiphany.dll
Requested dispatched. Request data length was O

Processing new request.
...
Request data length was 72
```

The log file consists of blocks, with each block representing a request to the proxy DLL. Each new request starts with a `Processing new request` message, and then a request header. The header is followed by the data section if the request was of the type `POST`. First, a number of bytes is printed (172), then the actual data. Finally, if the request was successful, the log entry ends with the following message:

```
Requested dispatched. Request data ....
```

If a request fails, the entry ends with the following message:

```
FAILED TO DISPATCH REQUEST DUE TO A SOCKET ERROR.
```

Proxy Redirection

The E.piphany proxy allows you to set up automatic redirection to another URL. This may be desirable when you would like to move users from an old AppServer host machine to another host. The AppServer need not be running on the old machine, so long as the proxy remains installed. To configure the proxy to redirect access to the URL for a new AppServer, enter the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Epiphany\Instances\instance\
RedirectURL
```

EXPORT/IMPORT OF METADATA

All of the control information for an EpiCenter datamart is stored in a metadata repository called EpiMeta. This repository takes the form of a transactional, relational database composed of numerous tables and an extensive set of declarative referential integrity constraints. E.piphany provides the EpiCenter Manager administrative interface for configuring this metadata.

E.piphany also provides a metadata Export/Import utility for backing up the definition of an EpiCenter and moving metadata from one EpiCenter to another. To use the Export/Import tool properly, you should understand the basic metadata concepts discussed in this appendix.

METADATA OVERVIEW

All of the user-configurable metadata tables have integer primary keys. These non-natural keys are provided by the database engine when a metadata row is inserted. The value of the primary key itself has no intrinsic meaning. All relationships to this inserted row are made via this integer. For example, the relationship between a dimension role and the base dimension from which it is derived is represented in EpiMeta by an integer column called `dim_base_key`.

The Access Export database does not simply contain a copy of the metadata tables being exported for this reason: If data were copied to the Export file, then when this same information is imported into a new EpiMeta, the integer primary key values in the Export file might clash with already existing primary keys in the target EpiMeta. Therefore, the Access database uses an EpiMeta-independent representation of metadata. See "Export File Format," on page 494 for a description of these Access database tables.

EpiMeta contains many tables, all of which are inter-related. In order to export only a portion of the metadata at a time, the Export machinery must decide where to stop exporting—otherwise, the entire metadata must be exported with each operation.

When using the **Export** command of EpiCenter Manager, you can select which part of the metadata to export. The Exporting Metadata dialog box is organized by related metadata: **Schema**, **Measures**, **Extraction**, **Presentation**, **Security**, **Storage**, and **Runtime**.

Reimporting objects does not delete references to those objects. For example, a measure definition can be reimported without deletion of the measure mappings in Web pages that point to that measure. If the Import machinery finds these references, then the same relationships are established in the new EpiMeta as the ones that were exported. However, if the Import machinery does not find these measures by name, the measure mapping information is lost upon import.

Replacing Existing Metadata on Import

When importing metadata into an existing EpiMeta, the import machinery detects an attempt to overwrite existing metadata. Existing top-level objects are overwritten in place. Their children are deleted, however, before being recreated. The definition of "existing" is usually based on a unique name column for one of the metadata tables. For instance, Web pages must have unique names, so an attempt to import a Web page with the same name as an existing one results in a warning message (unless the **Always Replace Existing Data** option has been selected previously).

TRANSACTION TYPES

The Transaction Types tab of the EpiCenter Configuration dialog box allows you to rename transaction types. However, unlike most other metadata, the name of the transaction type is used in dimension tables, not the key. This leads to problems when you try to export and then import metadata. If you rename an existing transaction type, that change in metadata is propagated tc the local metadata tables. When you export transaction-type definitions and then import them to another instance, EpiCenter Manager does not overwrite the existing transaction types–it adds new ones.

Data that references the old transaction type continues to do so after importing the new type. If you have renamed existing transaction types in the source system for your metadata, you must restore the original names of each transaction type before exporting metadata. E.piphany recommends that you use the `$$TRANSTYPE` macro whenever you refer to a transaction type in an SQL statement. Refer to Appendix A, "E.piphany Macros," for details.

ACTUALS AND AGGREGATE METADATA

The **Export All** command in EpiCenter Manager does not export the entire contents of EpiMeta. The options for Actuals and Aggregates are omitted by default from this export. This is because these sections of metadata are "derived" metadata: the Adaptive Schema Generator produces Actual table metadata, and Aggbuilder produces aggregate metadata. As a result, a new EpiCenter can be constructed using the Export All metadata by rebuilding the schema, re-extracting, and re-running Aggbuilder.

To "clone" an EpiMeta in such a way that EpiMart itself does not need any modification when it is used with this new EpiMeta, you must select the **Runtime** options for **Actuals** and **Aggs** in the Exporting Metadata dialog box.

Warning: *An Export All operation on a running system, followed by a re-import of that metadata causes all Actual and Aggregate metadata to be lost.*

Export File Format

Each Microsoft Access Export database has the same schema. This schema can be thought of as a metaschema for representing relational data. For a description of these tables, see Table 58.

To modify the row contents contained in an Export file, edit the **Export_col** table, which is simply a collection of name/value pairs for columns.

Table 58: Export Tables

Table Name	Description
Export_tbl	One row per metadata table being exported.
Export_row	One row per metadata row being exported.
Export_col	One row per column per row of metadata being exported. Only non-relationship columns are contained in this table.
Export_rel	One row per relationship between two rows of metadata. Can be a relationship between two rows contained in the Export file, or between one row in the Export file and one reference to a row in a foreign EpiMeta.
Export_status	Header information about the Export file.
Rel_parent	A reference to a metadata row in the foreign EpiMeta.

TROUBLESHOOTING

This chapter describes the E.piphany error conditions and error messages and suggests action you can take to resolve problems.

Warning: *Using Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk.*

For information about how to edit the Registry, view the "Changing Keys And Values" Help topic in Registry Editor (**Regedit.exe**), or the "Add and Delete Information in the Registry" and "Edit Registry Data" Help topics in **Regedt32.exe**.

Note: *E.piphany suggests that you back up the Registry before you edit it.*

Any number of problems can arise when either your AppServer host or the database-server host machine runs out of disk space. Whenever you experience unfamiliar or intermittent errors, check to ensure that your host machines have an adequate amount of free disk space.

If the corrective actions that are presented in an error message that the AppServer displays, or in the discussion of problems presented in this chapter, do not correct the problem, please send e-mail with a description of the problem and the steps you have taken to correct it to E.piphany Customer Support at `support@epiphany.com`.

APPLICATION SERVER ERROR MESSAGES

Note: *If the Application Server fails to start, you can run the Scrutiny debugging tool in EpiCenter Manager to diagnose the problem. See "Running the Scrutiny Debugging Tool," on page 299.*

The following error messages are described on the indicated pages:

- "User Cannot Log In," on page 496
- "Internal Windows NT Error," on page 498
- "EpiQuery Engine Database Connection Open Failure Exception," on page 498
- "Charts Do Not Display," on page 498
- "GIF Images Fail to Display on Web Pages," on page 499
- "Result Page Error: Extraction Date Unknown," on page 500
- "Web Server Message: Object Not Found," on page 500
- "Browser Crashes When Retrieving Results from Application Server," on page 502
- "Refresh Program Fails," on page 503
- "Application Log Full Error," on page 503

USER CANNOT LOG IN

There are three types of failure associated with an unsuccessful login to the Application Server:

1. The Application Server generates a critical exception indicating an application error, or an error related to the NT security domain).

   If this case, an error message appears in the browser with a link to the log that contains the stack trace for the error. Read the error description. Generally, it relates to the way NT domain security is configured or set up at this point. For example, the error might say that the domain controller could not be reached.

2. An invalid login message displays even though there is a valid username and password.

   This error may result when the search for the username/password occurs on the wrong machine. For example, suppose a user *xyz* existed on both the machine local to the Application Server and the primary domain controller.

   The order in which the user *xyz* is searched is as follows:

   - the local SAM
   - the primary domain
   - the trusted domains

   Thus, user *xyz* in the local SAM will be found first, even though the username/password combination could have been for the user *xyz* that exists in the primary domain or in a trusted domain.

   To solve this problem, further specify the username by including the domain name before the username; for example, *EPIPHANY\xyz*. In general, specify the full user's name if the user's browser displays an invalid login message.

3. Authentication was successful, but the user is not allowed to use the E.piphany system.

   For a user to have access to E.piphany System, he or she must be a member of at least one E.piphany group after a sync-up process.

   Use EpiCenter Manager to check group memberships after you receive this error message. If the old group memberships were removed, then this error has occurred because in the NT domain, the user is not a member of the NT groups that have the corresponding groups marked Synchronize in EpiCenter Manager.

   Also, check that the username used to log in matches the username in EpiCenter Manager. If the username in EpiCenter Manager is prefixed with domain name other than the one that the actual NT user is a member of, then the login could fail and return an error message to this effect. Specifying the full username upon log in may fix this problem.

INTERNAL WINDOWS NT ERROR

This error results when the E.piphany Application Server cannot be started as a service. The exact error message that was returned from the E.piphany Application Server is logged in the Windows NT Event Log.

To locate this error message in the Event Log:

1. Go to **Start\Programs\Administrative Tools (Common)\Event Viewer.**
2. Click the **Log** menu and select the Application.
3. Double-click the appropriate event.

   All E.piphany Application Server events have the source EpiAppServer.

To solve this problem, first stop any running Application Server services. Then log onto Windows NT as an administrator and re-install the E.piphany software. If this does not solve the problem, start the Application Server from the command line as described in "Monitoring the Application Server," on page 360. The console output should describe what is wrong.

EPIQUERY ENGINE DATABASE CONNECTION OPEN FAILURE EXCEPTION

An **EpiQueryEngineDBConnOpenFailureException** from the EpiQueryEngine means that the Application Server is having difficulty connecting to the datamart. Take these steps to correct this problem:

- Make sure that the username and password for the EpiMeta database are set correctly in the Windows Registry.
- Make sure that the database name and server are also configured properly (again, in the Registry).
- Make sure that TCP/IP connectivity is properly configured.:

CHARTS DO NOT DISPLAY

If there is a broken link to an image, or no image appears, or an image with an error message appears instead of the chart, there is a charting problem. Please contact E.piphany Customer Support.

GIF IMAGES FAIL TO DISPLAY ON WEB PAGES

If GIF images do not appear on your Web pages, do the following:

1. Make sure that your Web server has an alias for your *instance_name* that points to a valid directory. If you performed a normal installation, then all Web files should be located in the directory:

   ```
   C:\Program Files\Epiphany\instance_name\web\WWWROOT
   ```

   and GIF files should be located in the **images** subdirectory:

   If your Web server is IIS 3.0, you can find the aliases by opening up the IIS Internet Service Manager (normally this is located in your **Start\Microsoft Internet Information Server** menu) and selecting the Directory tab. Make sure that there is an entry for *instance_name* that has a valid directory.

   For the IIS 4.0 Web server, open the Windows NT **Option Pack\Microsoft Internet Server\Internet Service Manager**. Aliases are listed in the IIS Management Console.

2. Make sure the **WWWROOT\images** directory has the GIFs in it. If you are missing a few GIFs, then you need to reinstall your Application Server or copy the GIFs from a different instance.

3. Check the BASE HREF tag that is defined in the source of the page. In your browser, try to view the source for this HTML page. Look for the BASE HREF tag at the top of the page. Note what it is and make sure that it is a valid alias using the procedure above.

4. Make sure your Web server is serving pages and that your browser is not displaying cached HTML. Clear the caches (Memory and Disk) on your browser, close your browser, and try to access the URL again. Also, try referencing another URL from your Web server to make sure that it is running.

   Technical Note: All of the GIFs on an Application Server-generated page are referenced from the **images** directory, which is relative to the BASE HREF specified at the top of the page in a META tag. The *instance_name* is derived from the URL that you use to access the Application Server. It is used throughout the system to read the correct Registry entries and to generate the correct URLs.

Result Page Error: Extraction Date Unknown

The `last_extract_date` is a field that is kept in the EpiMeta database. It is used to keep track of the date displayed on the top of all reports as the date of the last extraction. It is normally populated by the extraction SQL entered in the End of Extraction extraction group in the EpiCenter Manager. It can also be populated by EpiCenter Manager via the Configuration dialog box. This field must be entered in one of two very strict formats. The default format is `mm/dd/yyyy`; for example: 01/14/1998.

The Application Server applies the following logic to parse the date:

1. If the field has more than 10 characters, then parse it using the pattern `mm/dd/yyyy hh:mm:ss`. Otherwise, use the pattern `mm/dd/yyyy`.
2. If the parse fails, then use `{extraction date unknown}`.

In addition, the date that is displayed at the top of the report always has a time zone. The time zone is printed based on the default time zone of the machine on which the Application Server is running. The date that is being displayed is also taken into consideration. For example, the date 12/20/1997 displays as **December 20, 1997 PST** if the Application Server was running on a machine in California. However, the day 05/12/1998 will display as **May 12, 1998 PDT** on the same machine since Daylight Savings time took affect in April.

Note: EpiCenter Manager does not allow the user to enter a date in the format `mm/dd/yyyy hh:mm:ss`. Only the SQL in the extraction job can enter dates of this format, or you can manually arrange this via ***isql.***

Web Server Message: Object Not Found

If you installed the E.piphany software using the E.piphany installer, you can access your Web server through this type of URL:

```
http://hostname/scripts/instance_name/Epiphany.dll
```

If you receive an object not found error message, follow these steps:

1. Start and stop the Web server. If this does not solve the problem, go to the next step.
2. Verify the Web server is serving pages.

Note: *Make sure that your browser is not just returning cached HTML pages by clearing your memory and disk cache before iesting.*

Try to access other URLs from the same *machinename*. Try to access other static HTML files that are installed as a part of the Application Server installation, such as:

```
http://machinename/instance_name/WWWRoot/help/WebPages.html.
```

3. If this does not work, try accessing any other file that the Web server should be serving. Consult the Internet Service Manager for the names of other aliases that the Web server should be serving, and then try to access these aliases with your browser.

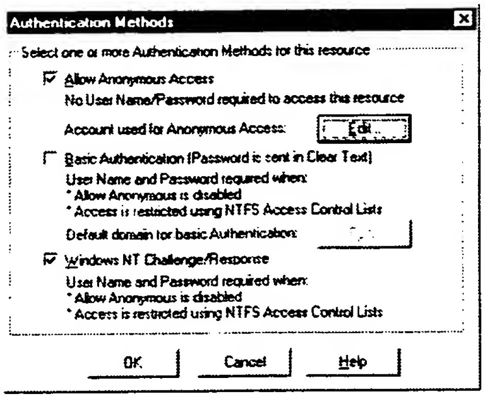
   In most cases, your Web server searches the **C:\Inetpub\scripts\*instance_name*** directory to find the **Epiphany.dll** program. Make sure that there is such a directory on your machine, and that the **Epiphany.dll** file is in that directory.

4. Check the file permissions for the **Epiphany.dll** file.

   First, make sure that the account IIS uses for anonymous logins has file access permissions for the **Epiphany.dll** file. Go to the IIS 3.0 Internet Service Manager and look at the Anonymous Login account box. In IIS 4.0, right-click the name of the machine and choose Properties. Select the Directory Security tab.

In the Authentication Methods dialog box, select **Allow Anonymous Access** and click the **Edit** button to modify the account used for this purpose. Make sure the user and password are correct.

FIGURE 116: AUTHENTICATION METHODS DIALOG BOX



Authentication Methods

Select one or more Authentication Methods for this resource

☑ Allow Anonymous Access

No User Name/Password required to access this resource

Account used for Anonymous Access:

□ Basic Authentication (Password is sent in Clear Text)

User Name and Password required when:
* Allow Anonymous is disabled
* Access is restricted using NTFS Access Control Lists

Default domain for basic Authentication:

☑ Windows NT Challenge/Response

User Name and Password required when:
* Allow Anonymous is disabled
* Access is restricted using NTFS Access Control Lists

To check file permissions, open the Windows NT Explorer window and right-click the **Epiphany.dll** file in the **./scripts/*instance_name*** directory specified above. Go to Properties and open the Security tab. Click **View Permissions** and make sure that the account that you used for Anonymous Web Login has access to this file. (If **Everyone** is selected, then all people, including the Web login account, have access to the file.)

BROWSER CRASHES WHEN RETRIEVING RESULTS FROM APPLICATION SERVER

In general, for machines with less than 32 megabytes of RAM, the browser will perform very poorly when parsing HTML files that are larger than 150 kilobytes. Therefore, users who do not have at lease 32 megabytes of RAM installed on their machines sold refrain from retrieving large queries.

An example of a large query follows. Suppose you query Customer by Fiscal Year and apply the filter Business Unit: Learning. This query returns approximately 3800 customers, and the HTML that is generated is 1.8 megabytes. When loaded in Netscape 4.03, it occupies 37 megabytes, takes 3 minutes to parse, and consumes the entire processor. The parsing is the bottleneck in the downloading of the file.

Note: *When started, Netscape Navigator 4.03 requires 8 megabytes immediately to load whatever it is loading. Microsoft Internet Explorer (IE) requires 6.7 megabytes, and is substantially faster at parsing the text.*

REFRESH PROGRAM FAILS

If the Refresh program fails, the AppServer continues to use the old metadata information. Users will still be able to log in and view the old Web Pages. This is because the Refresh program reads the new metadata into temporary structures that are atomically exchanged with the old structures only if all of the refresh structures were read correctly.

To solve this problem, rerun the **Refresh** program.

APPLICATION LOG FULL ERROR

If you receive this error, you can take the following action to delete all log events.

Note: *If you do not want to delete all of the log events, you can also increase the amount of space allocated to the application log. You can also select the Override as Needed option in the Log Properties dialog box.*

1. Open **Start\Programs\Administrative Tools (Common)\Event Viewer.** Select the **Application Log** under the Log menu item.
2. Select **Clear All Events** from the Log menu.

   When prompted whether you want to save the log files, and whether you really want to delete all of the log events, respond in the affirmative.

EPICENTER MANAGER ERROR CONDITIONS

EpiCenter Manager operations can fail to complete when the disk volume on which the EpiMart database resides becomes full. The AppServer, in addition to its log files, maintains a record of database queries in metadata. The table that contains these records is used by the aggregate optimizer to select candidate tables and columns for aggregates. If this table grows to the point at which it fills the disk volume, requests to alter the metadata that you make with EpiCenter Manager can fail with a variety of error messages from your database server.

Refer to "Tracking of Queries for Aggregate Optimization," on page 371 for information on how to resolve this problem.

GLOSSARY

A

Actual Tables

The Actual tables are metadata created by the Adaptive Schema Generator after it has built or modified tables in EpiMart.

These tables are consulted after a schema generation to determine delta operations to perform. Other tools examine them to ensure that the EpiMart schema matches the current metadata definition.

Ad Hoc Query

An improvised query that is composed and executed as the direct result of a user request rather than being scheduled with other batch operations.

Aggregate

An aggregate is a pre-computed summary of an EpiMart table that is designed to improve query response time.

Aggregate Builder (Aggbuilder)

Aggbuilder is the executable program (**agg.exe**) that builds aggregate tables in EpiMart.

Normally, the execution of this program is scheduled after all data has been extracted from source systems and merged into EpiMart.

Aggregate Instruction

An aggregate instruction is a specification of a single aggregate to be built on a fact table. The aggregate builder executable is used to build the aggregate specified by an aggregate instruction. 165

Aggregate Navigation

The process by which the E.piphany query machinery determines the optimal aggregate table for addressing an end-user query.

Aggregate Optimizer

The Aggregate Optimizer is a tool that helps database administrators construct optimized sets of aggregates based on user-query logs.

Application Server

The Windows NT Service that connects with a Web server and serves up E.piphany Web pages and reports. An Application Server is configured to connect with an EpiCenter.

Approximate Count

A statistical method for estimating the number of members in a list, which takes far less time than counting.

AppServer

See Applicaton Server.

AppServer Monitor

A Web-based interface for monitoring and refreshing the Application Server.

Attribute

Attributes, which are derived from columns in dimension tables, appear as items that end users can select on Web pages. Attributes also serve as filters that users apply to results to refine (drill down) their queries.

Attribute Factory

An EpiCenter Manager tool for creating tansaction filter filters (two-occurrence filtering). Filters are stored as attributes, so creating a new filter creates a new attribute, hence the term Attribute Factory.

B

Backfeed Table

A backfeed table is an EpiMart table that holds data about campaigns that have been run. Data can be extracted from a backfeed table for use in the EpiCenter.

Backlog Type

The backlog type indicates that a measure term is to display accumulated data. BEGIN indicates accumulation through the start of a reporting period. END indicates accumulation through the end.

Base Dimension

A base dimension is a physical dimension table in EpiMart. Base dimension tables contain the actual dimension values. Base dimensions are defined globally across an EpiCenter.

Base Dimension Aggregate

A base dimension aggregate is a table in EpiMart that represents an aggregated view of a base dimension table.

Base Table

Base tables are the EpiMart tables (both fact and dimension) that store nonaggregated data at extraction granularity. They are used as input to Aggbuilder to build aggregate tables. Base table names end in the _0 suffix.

Behavior Component

One portion of the specification for a link behavior. A link component indicates whether or not to carry state or automatically create a report.

C

Calendar

A Web page type that displays scheduled campaigns.

Campaign

A campaign is the specification for a set of output files that each contain a treatment code and a list of recipients. When a campaign is run, a query creates and populates the data files.

Campaign Manager

An E.piphany module that captures data related to your current marketing program. Reported items include campaign name, expected revenue, type of mailing, and so forth. When you send targeted mail, you have run a campaign.

Carry State

The behavior component that specifies that a link carries state information forward into the Web page at the destination node.

Clusters

Clusters are natural groupings that occur within your data. The Community Clusters Web page type identifies such clusters. *See also,* Fact Clusters.

Community Clusters

A Web page type that identifies groupings within sets of attributes.

Cumulative Projections

A Web page type that projects cumulative results for a current time period based on previous time periods and current results so far.

Current Table

An EpiMart fact table that contains recently-extracted data. Current tables have suffixes of _A or _B.

Custom Fact Index

A custom fact index defines indexes, in addition to Date, to build for a fact table.

D

Data Store

A data store is a logical location from which data is extracted to, or downloaded from, an EpiCenter. Data stores include relational database connections, files, and directories.

Data Store Role

In an extraction job, a data store role specifies how a selected data store is to be used in the job.

Data Warehouse

A data warehouse transforms the raw data from an organization's source-system databases into a star-schema format that is optimized for end-user ad hoc query and analysis.

Database Schema

A database schema is a structure for organizing source data into tables through which information can be accessed by end users in a concise format, such as reports, charts, lists, and campaigns.

Date Dimension

The Date dimension is a special base dimension table supplied by E.piphany that stores all attributes related to time. All fact tables receive the **date_key** foriegn key into this table.

Degenerate Dimension

A degenerate dimension is a textual field, such as a bill of lading number, stored directly in the fact table, as opposed to standard dimensions, for which the fact tables store foreign keys rather than the data itself.

Demographic Base Dimensions

The demographic base dimension tables for individuals and groups are used by the List Manager and Campaign Manager to generate lists of individuals and groups.

Demographic Dimensions

Individual and group dimension roles associated with the **Ind_Group_Joiner** table.

Demographic Filters

Filters that are based on attributes of the **group** or **indiv** dimension roles.

Dependency

A dependency is a relationship between two queues that prevents them from being executed at the same time. If Queue B depends on Queue A and both queues are ready to be executed, then the scheduler executes Queue A before Queue B.

Dimension Column

A dimension column is a single column or attribute of a base dimension table. It contains the actual customer data that appears on reports or in filters.

Dimension Column Access

Dimension column access is the ability for a user or group to view only a subset of the available data in an EpiCenter. Dimension-column-access settings can limit specific users to data for only one region.

Dimension Column Set

A dimension column set is a named set of dimension columns (all within a single base dimension) that defines which columns are used in a base dimension aggregate.

Dimension Role

A dimension role is a logical view of a base dimension table that allows data from that table to appear in reports of differing scope.

Dimension Tables

Dimension tables contain attribute data, such as the names of customers and territories. Attributes are represented as global objects in an EpiCenter.

E

E.piphany Application

An E.piphany application is a set of one or more Web pages and a topic master that provides a map for navigating among those pages to solve a class of business problems.

E.piphany Solution

An E.piphany solution is an instance of an E.piphany application that has been installed and configured for a particular organization. 32

EpiCenter

An EpiCenter is a single logical E.piphany datamart that includes customer data and control metadata. It physically consists of two databases: EpiMeta and its associated EpiMart.

EpiCenter Manager

E.piphany's versatile program for configuring, administering, re-configuring, and updating your database. You can use its graphical user interface to define your EpiMeta database table schema.

EpiCenter Manager is also the means by which you configure Web pages for end users, set up the jobs needed for data extraction, and perform administrative tasks.

EpiChannel

EpiChannel is the E.piphany program (**extract.exe**) that executes extraction jobs. EpiChannel implements the extraction steps that are stored as metadata in the EpiCenter.

EpiMart

EpiMart is the physical database that contains actual datamart data. The schema of tables in EpiMart is determined by the metadata in EpiMeta. The data contents of EpiMart tables are determined by the extraction steps in EpiMeta.

EpiMeta

EpiMeta is the metadata repository for an EpiCenter. It contains all control information for the EpiCenter and therefore defines the behavior of that EpiCenter. EpiMeta points to EpiMart via the special EpiMart data store.

External Table

An external table is a table in any data store that is made available for use by the E.piphany system. External tables are commonly used as intermediate tables in multi-staged extractions.

Extraction

Extraction is the process of copying data from external source systems into an E.piphany EpiCenter.

Extraction Group

An extraction group is an ordered collection of extraction steps.

Extraction Step

An extraction step is a single atomic extraction operation. It can be an SQL statement, a semantic instance, a truncation step, an extraction group, or a system call.

F

Fact Aggregate

A fact aggregate is a physical table in EpiMart that contains aggregated fact information for a single fact table.

Fact Cluster

A fact cluster is a user-defined index into a list manager fact table based on a selected dimension. Fact clusters can significantly improve list manager performance.

Fact Column

A fact column is a single numeric column in a fact table.

Fact Count

A fact count is a special index on list manager tables that allows Oracle database servers to make use of fact clusters.

Fact Table

A fact table is a physical table in EpiMart that contains numeric data and references to dimension tables that specify attributes about that data. Facts are represented as global objects in an EpiCenter.

Fact-Table Cluster

A copy of a fact table that is sorted on an attribute.

Fact-Table Counts

A record of how often entries for a group or individual appear in a fact table.

Filter Element

A filter element is a single check box or entry within a list box that appears on a Web page.

Filter Group

A filter group is a logical grouping of filter elements.

Filters, Attributes, Objects, and Measures

These are the parameters that a user specifies in a Web page to create a typical report.

FOAM

See Filters, Attributes, Objects, and Measures. 303

Foriegn Key

A reference in one relational-database table to a key value that resides in another table.

G

Glossary Entry

A glossary entry provides end users with online definitions for configured elements that appear on Web pages.

Group Campaigns

A Web page type that creates campaigns directed toward groups.

Group Dimenson

a special dimension that stores demographic data about groups for lists and campaigns.

Group List Manager

A Web page type used to create lists of groups such as compaines or households.

H

Help Page

A Web page that displays usage information about E.piphany Web pages or general information about E.piphany solutions.

High/Low Clusters

A Web page type that identifies combinations of attribute values associated with highest and lowest values.

History Table

An EpiMart fact table that contains older data that is expected to change infrequently. Historical tables have suffixes of _X or _Y.

Home Page

A Web page that is displayed first when a user logs in.

I

Ind_Group_Joiner

This is a special fact table for lists and campaigns that connects the individual and group base dimension tables.

Individual Campaigns

A Web page type that creates campaigns directed toward individuals.

Individual Dimension

A special dimensions that stores demographic data about individuals for lists and campaigns.

Individual List Manager

A Web page type that is used to create lists of individuals.

Influences

A Web page type that is used to identify predictive relationships among a set of attributes.

Intermediate Web Page Types

Web page types that allow users to fill in missing state information before jumping to a node that requires that state in order to automatically generate a report.

J

Job

A job is a top-level workflow object that defines a sequence of data extraction steps to be performed by EpiChannel. It also specifies the data stores for that execution.

Job Step

A job step is a single unit of execution in an extraction job.

L

Lifecycles

A Web page type that projects the life cycle for a new product based on previous like products.

Link

A hypertext link that allows a user to jump from one E.piphany Web page to another. Links have associated behaviors that determine if state is to be carried forward and if a report is to be generated automatically.

Link Category

A set of links that appear together on a Web page. The link category determines where a link appears on the Web page at the originating node.

List

A list is a set of demographic row keys that is created by an end-user query and is saved in the Report Gallery.

List Manager

An E.phiphany module that allows end users to generate lists related to individual and group demographic data. Other Web pages can utilize lists as filters.

Login Page

A Web page displayed by the AppServer that allows users to confirm their identities before gaining access to an E.piphany application.

M

Measure

A measure is a single business calculation. It can be an arithmetic combination of measure terms using RPN (Reverse Polish Notation).

Measure Mapping

The mapping of measures to Web page selections.

Measure Set

A measure set consists of one or more measures used together (as a set) that apply to an Influences or Community Clustering Web page. Measure sets are associated with one of three types: Classification, Regression, or Clustering.

Measure Term

A measure term is a single component of a measure. It can be combined with other measure terms to create a composite measure (business calculation).

Modeling

A Web page type that identifies predictive relationships for use in scored lists.

MomentumBuilder

MomentumBuilder is the executable (**momentumbuilder.exe**) that creates the special tables that are used to generate lists and campaigns.

N

Navigation Node

A navigation node is a location within a topic to which a Web page is assigned. A user performs an activity in the page at that node, then follows links to other nodes.

Navigation Type

The link-behaivor component that indicates the window in which the Web page at a destination node is to be displayed. This window could be the same as the origin, a pop-up, or another browser window.

News Banner

An area of the Home page that you can customize to display a daily message or an applet.

O

Object Gallery Pane

The Object Gallery is a pane in an EpiCenter Manager dialog that allows you to drag available objects and drop them into the appropriate configuration categories.

P

Primary Dimension

A primary dimension is a dimension that is the target of a data-mining analysis.

Profiling

A Web page type that charts comparative values.

Proxy

An ISAPI application that mediates I/O between the Web server and the E.piphany Application Server.

Pull/Push SQL Statement

A pull/push SQL statement is a type of extraction step that issues SQL against an input data store and then pushes the result set into a table in the output data store.

Q

Query Machinery

The component of the E.piphany Application Server that communicates with EpiMart and issues SQL statements against the DBMS.

Queryable Column

A queryable fact or dimension table column is a column that can be used in a List Manager or Campaign Manager query.

Queue

A queue is a collection of tasks that can be executed by the scheduler. The scheduler executes queues according to their dependencies. Within a queue, tasks are executed sequentially based on their assigned priority level.

R

Report

A report is a saved set of Web-page settings. Users can save and retrieve reports in the Report Gallery. Saved reports do not include results.

Report Gallery

A component of E.piphany's front-end user interface that allows users to save and retrieve reports, lists, and campaigns. 165

Rows and Columns

The Basic Rows and Columns Web page type displays measure data that is classified across a pair of attributes. Advanced Rows and Columns allows nested drill-down reports.

S

Schedule

A Web page type that allows users to schedule campaigns.

Scheduler

The Scheduler is the E.piphany sybsystem that schedules and executes automated tasks, such as extraction jobs and campaigns.

Schema Generator

The Schema Generator is a component of EpiCenter Manager that builds the tables in EpiMart using metadata definitions in EpiMeta.

Scoring

A Web page type that allows users to rank the members of a list according to a predictive model or a measurement.

Scrutiny

Scrutiny is a debugging tool that identifies inconsistencies in the metadata that specifies your EpiCenter configuration.

Seed

A seed is a predetermined recipient that is used to validate campaigns. The IndivSeed and GroupSeed tables can contain entries for seeds.

SeedJoiner Fact Table

The SeedJoiner fact table is a built-in fact table that must be populated to list all seed records.

Select a Navigation Path Web Page Type

A Web page type that allows users to choose from among several links, but does not execute a report.

Semantic Instance

A semantic instance is an SQL program that applies business rules to extracted data. It is an instantiation of a semantic type.

Semantic Template

A semantic template is an SQL template for a semantic type.. When the template is instantiated as a semantic instance, the resulting SQL refers to actual columns and tables.

Semantic Type

A semantic type is a logical business process that is implemented in a semantic template.

Solution Map

A solution map is a typical series of steps that a user might take to solve a particular type of business problem. *See also* Topic Master.

Source System

A source system is a repository of business data, such as an OLTP system, from which data is extracted into an EpiCenter datamart.

SQL Statement

A SQL statement is an extraction step that issues custom SQL against an input data store. The results of the SQL statement can either be discarded or used to push data into a table in the output data store.

Staging Table

A staging table is an intermediate table in an EpiCenter that holds source system data in preparation for merging into the datamart.

Stand-Alone SQL Statement

A stand-alone SQL statement is a SQL statement whose results are discarded.

Star Schemas

A dimensional data warehouse uses a star schema to organize tables. At the center of a standard star schema is a fact table that contains measure data.

Radiating outward from the fact table like the points of a star are multiple dimension tables that contain attributes. In an E.piphany star schema, the stars are interjoined.

State Information

The accumulated set of reporting parameters that have been selected in current and previous Web pages, which can be carried forward to the Web page at a destination node.

System Call

A system call is an extraction step that causes an operating system program to be invoked.

T

Tab

A screen or area within an EpiCenter Manager dialog box that is accessed by clicking the image of a tab near the top of the dialog box.

Task

A task is an activity that can be scheduled in a queue. Tasks include such things as extraction jobs and campaigns.

Task instance

A task instance is a single scheduled or completed execution of a task.

Topic

A topic is a named set of navigation nodes, the Web pages assigned to each node, the links between each node, and the behaviors associated with each link.

Topic Master

A template for a topic that includes a predefined set of navigation nodes, links, and behaviors but omits specific Web pages.

Transaction

An action that an individual or a member of a group might participate in, such a purchase, a return, a call to a call center, the receipt of a treatment, or a response to a campaign.

Transaction Filter

A transaction filter allows a user to filter dimension demographics by measure data. Transaction filters are associated with a fact table, a transaction type, and a set of attribute filters.

Transaction Filter Filters

A transaction filter filter, which is defined via EpiCenter Manager's Attribute Factory allows two-occurence filtering.

Transaction Type

Transaction types distinguish rows with different interpretations in the same fact table. For example, an Order fact table might contain both a BOOK and SHIP transaction type.

Transaction Type Set

Multiple transactions can be grouped as a set and applied to a single measure term. Applying a transaction type set allows the use of a single calculation (usually SUM) for all rows for all transaction types in the set.

Trends

A Web page type that identifies trends along a set of attributes and projects expected values based on those trends.

U

UNKNOWN Dimension Row

The **UNKNOWN** dimension row is a special row in every dimension table that is used as a referent by all fact table rows that do not refer to an actual dimension element.

By convention, the string UNKNOWN is generally used to refer to this dimension row.

V

View List

A Web page type that allows users to view and download data pertaining to the current memberhsip of a list.

W

Web Page

A component of the E.piphany user interface that is displayed by a Web browser. An E.piphany Web page allows a user to request reports, lists, and campaigns, and to navigate to other Web pages.

Web Page Type

A prototype for a Web page that specifies the overall appearance of the page, the types of reports that a user can request, and the types of reusable data elements that can appear.

INDEX